

Optimization

of chemical processes
second edition

Edgar • Himmelblau • Lasdon

OPTIMIZATION OF CHEMICAL PROCESSES

McGraw-Hill Chemical Engineering Series

EDITORIAL ADVISORY BOARD

Eduardo D. Glandt, Professor of Chemical Engineering, University of Pennsylvania

Michael T. Klein, Professor of Chemical Engineering, Rutgers University

Thomas F. Edgar, Professor of Chemical Engineering, University of Texas at Austin

Bailey and Ollis: *Biochemical Engineering Fundamentals*

Bennett and Myers: *Momentum, Heat, and Mass Transfer*

Coughanowr: *Process Systems Analysis and Control*

deNevers: *Air Pollution Control Engineering*

deNevers: *Fluid Mechanics for Chemical Engineers*

Douglas: *Conceptual Design of Chemical Processes*

Edgar, Himmelblau, and Lasdon: *Optimization of Chemical Processes*

Gates, Katzer, and Schuit: *Chemistry of Catalytic Processes*

King: *Separation Processes*

Luyben: *Essentials of Process Control*

Luyben: *Process Modeling, Simulation, and Control for Chemical Engineers*

Marlin: *Process Control: Designing Processes and Control Systems for Dynamic Performance*

McCabe, Smith, and Harriott: *Unit Operations of Chemical Engineering*

Middleman and Hochberg: *Process Engineering Analysis in Semiconductor Device Fabrication*

Perry and Green: *Perry's Chemical Engineers' Handbook*

Peters and Timmerhaus: *Plant Design and Economics for Chemical Engineers*

Reid, Prausnitz, and Poling: *Properties of Gases and Liquids*

Smith, Van Ness, and Abbott: *Introduction to Chemical Engineering Thermodynamics*

Treybal: *Mass Transfer Operations*

McGraw-Hill Higher Education 

A Division of The McGraw-Hill Companies

OPTIMIZATION OF CHEMICAL PROCESSES, SECOND EDITION

Published by McGraw-Hill, a business unit of The McGraw-Hill Companies, Inc., 1221 Avenue of the Americas, New York, NY 10020. Copyright © 2001, 1988 by The McGraw-Hill Companies, Inc. All rights reserved. No part of this publication may be reproduced or distributed in any form or by any means, or stored in a database or retrieval system, without the prior written consent of The McGraw-Hill Companies, Inc., including, but not limited to, in any network or other electronic storage or transmission, or broadcast for distance learning.

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

This book is printed on acid-free paper.

1 2 3 4 5 6 7 8 9 0 DOC/DOC 0 9 8 7 6 5 4 3 2 1 0

ISBN 0-07-039359-1

Publisher: *Thomas E. Casson*

Executive editor: *Eric M. Munson*

Editorial coordinator: *Zuzanna Borciuch*

Senior marketing manager: *John Wannemacher*

Project manager: *Vicki Krug*

Media technology senior producer: *Phillip Meek*

Senior production supervisor: *Sandra Hahn*

Coordinator of freelance design: *Michelle D. Whitaker*

Cover designer: *JoAnne Schopler*

Cover image: *Corbis*

Supplement producer: *Jodi K. Banowetz*

Compositor: *Lachina Publishing Services*

Typeface: *10.5/12 Times Roman*

Printer: *R. R. Donnelley & Sons Company/Crawfordsville, IN*

Library of Congress Cataloging-in-Publication Data

Edgar, Thomas F.

Optimization of chemical processes / Thomas F. Edgar, David M. Himmelblau,
Leon S. Lasdon.—2nd ed.

p. cm.—(McGraw-Hill chemical engineering series.)

Includes bibliographical references and index.

ISBN 0-07-039359-1

1. Chemical processes. 2. Mathematical optimization. I. Himmelblau, David Mautner,
1923– . II. Lasdon, Leon S., 1939– . III. Title. IV. Series.

TP155.7 .E34 2001

660'.28—dc21

00-062468

CIP

CONTENTS

Preface	xi
About the Authors	xiv

PART I Problem Formulation

1 The Nature and Organization of Optimization Problems	3
1.1 What Optimization Is All About	4
1.2 Why Optimize?	4
1.3 Scope and Hierarchy of Optimization	5
1.4 Examples of Applications of Optimization	8
1.5 The Essential Features of Optimization Problems	14
1.6 General Procedure for Solving Optimization Problems	18
1.7 Obstacles to Optimization	26
References	27
Supplementary References	27
Problems	28
2 Developing Models for Optimization	37
2.1 Classification of Models	41
2.2 How to Build a Model	46
2.3 Selecting Functions to Fit Empirical Data	48
2.3.1 <i>How to Determine the Form of a Model</i> / 2.3.2 <i>Fitting Models by Least Squares</i>	
2.4 Factorial Experimental Designs	62
2.5 Degrees of Freedom	66
2.6 Examples of Inequality and Equality Constraints in Models	69
References	73
Supplementary References	73
Problems	74

3	Formulation of the Objective Function	83
3.1	Economic Objective Functions	84
3.2	The Time Value of Money in Objective Functions	91
3.3	Measures of Profitability	100
	References	104
	Supplementary References	104
	Problems	105

Part II Optimization Theory and Methods

4	Basic Concepts of Optimization	113
4.1	Continuity of Functions	114
4.2	NLP Problem Statement	118
4.3	Convexity and Its Applications	121
4.4	Interpretation of the Objective Function in Terms of its Quadratic Approximation	131
4.5	Necessary and Sufficient Conditions for an Extremum of an Unconstrained Function	135
	References	142
	Supplementary References	142
	Problems	142
5	Optimization of Unconstrained Functions: One-Dimensional Search	152
5.1	Numerical Methods for Optimizing a Function of One Variable	155
5.2	Scanning and Bracketing Procedures	156
5.3	Newton and Quasi-Newton Methods of Unidimensional Search	157
	<i>5.3.1 Newton's Method / 5.3.2 Finite Difference Approximations to Derivatives / 5.3.3 Quasi-Newton Method</i>	
5.4	Polynomial Approximation Methods	166
	<i>5.4.1 Quadratic Interpolation / 5.4.2 Cubic Interpolation</i>	
5.5	How One-Dimensional Search Is Applied in a Multidimensional Problem	173
5.6	Evaluation of Unidimensional Search Methods	176
	References	176
	Supplementary References	177
	Problems	177
6	Unconstrained Multivariable Optimization	181
6.1	Methods Using Function Values Only	183
	<i>6.1.1 Random Search / 6.1.2 Grid Search / 6.1.3 Univariate Search / 6.1.4 Simplex Search Method / 6.1.5 Conjugate Search Directions / 6.1.6 Summary</i>	
6.2	Methods That Use First Derivatives	189
	<i>6.2.1 Steepest Descent / 6.2.2 Conjugate Gradient Methods</i>	

6.3	Newton's Method	197
	<i>6.3.1 Forcing the Hessian Matrix to Be Positive-Definite /</i>	
	<i>6.3.2 Movement in the Search Direction / 6.3.3 Termination /</i>	
	<i>6.3.4 Safeguarded Newton's Method / 6.3.5 Computation of</i>	
	<i>Derivatives</i>	
6.4	Quasi-Newton Methods	208
	References	210
	Supplementary References	211
	Problems	211
7	Linear Programming (LP) and Applications	222
7.1	Geometry of Linear Programs	223
7.2	Basic Linear Programming Definitions and Results	227
7.3	Simplex Algorithm	233
7.4	Barrier Methods	242
7.5	Sensitivity Analysis	242
7.6	Linear Mixed Integer Programs	243
7.7	LP Software	243
7.8	A Transportation Problem Using the EXCEL Solver	
	Spreadsheet Formulation	245
7.9	Network Flow and Assignment Problems	252
	References	253
	Supplementary References	253
	Problems	254
8	Nonlinear Programming with Constraints	264
8.1	Direct Substitution	265
8.2	First-Order Necessary Conditions for a Local Extremum	267
	<i>8.2.1 Problems Containing Only Equality Constraints /</i>	
	<i>8.2.2 Problems Containing Only Inequality Constraints /</i>	
	<i>8.2.3 Problems Containing both Equality and Inequality</i>	
	<i>Constraints</i>	
8.3	Quadratic Programming	284
8.4	Penalty, Barrier, and Augmented Lagrangian Methods	285
8.5	Successive Linear Programming	293
	<i>8.5.1 Penalty Successive Linear Programming</i>	
8.6	Successive Quadratic Programming	302
8.7	The Generalized Reduced Gradient Method	306
8.8	Relative Advantages and Disadvantages of NLP Methods	318
8.9	Available NLP Software	319
	<i>8.9.1 Optimizers for Stand-Alone Operation or Embedded</i>	
	<i>Applications / 8.9.2 Spreadsheet Optimizers / 8.9.3 Algebraic</i>	
	<i>Modeling Systems</i>	
8.10	Using NLP Software	323
	<i>8.10.1 Evaluation of Derivatives: Issues and Problems /</i>	
	<i>8.10.2 What to Do When an NLP Algorithm Is Not "Working"</i>	

References	328
Supplementary References	329
Problems	329
9 Mixed-Integer Programming	351
9.1 Problem Formulation	352
9.2 Branch-and-Bound Methods Using LP Relaxations	354
9.3 Solving MINLP Problems Using Branch-and-Bound Methods	361
9.4 Solving MINLPs Using Outer Approximation	369
9.5 Other Decomposition Approaches for MINLP	370
9.6 Disjunctive Programming	371
References	372
Supplementary References	373
Problems	374
10 Global Optimization for Problems with Continuous and Discrete Variables	381
10.1 Methods for Global Optimization	382
10.2 Smoothing Optimization Problems	384
10.3 Branch-and-Bound Methods	385
10.4 Multistart Methods	388
10.5 Heuristic Search Methods	389
10.5.1 Heuristic Search / 10.5.2 Tabu Search / 10.5.3 Simulated Annealing / 10.5.4 Genetic and Evolutionary Algorithms / 10.5.5 Using the Evolutionary Algorithm in the Premium Excel Solver / 10.5.6 Scatter Search	
10.6 Other Software for Global Optimization	411
References	412
Supplementary References	413
 Part III Applications of Optimization <hr/>	
11 Heat Transfer and Energy Conservation	417
Example 11.1 Optimizing Recovery of Waste Heat	419
Example 11.2 Optimal Shell-and-Tube Heat Exchanger Design	422
Example 11.3 Optimization of a Multi-Effect Evaporator	430
Example 11.4 Boiler/Turbo-Generator System Optimization	435
References	438
Supplementary References	439
12 Separation Processes	441
Example 12.1 Optimal Design and Operation of a Conventional Staged-Distillation Column	443

Example 12.2	Optimization of Flow Rates in a Liquid–Liquid Extraction Column	448
Example 12.3	Fitting Vapor–Liquid Equilibrium Data Via Nonlinear Regression	451
Example 12.4	Determination of the Optimal Reflux Ratio for a Staged-Distillation Column	453
	References	458
	Supplementary References	458
13	Fluid Flow Systems	460
Example 13.1	Optimal Pipe Diameter	461
Example 13.2	Minimum Work of Compression	464
Example 13.3	Economic Operation of a Fixed-Bed Filter	466
Example 13.4	Optimal Design of a Gas Transmission Network	469
	References	478
	Supplementary References	478
14	Chemical Reactor Design and Operation	480
Example 14.1	Optimization of a Thermal Cracker Via Linear Programming	484
Example 14.2	Optimal Design of an Ammonia Reactor	488
Example 14.3	Solution of an Alkylation Process by Sequential Quadratic Programming	492
Example 14.4	Predicting Protein Folding	495
Example 14.5	Optimization of Low-Pressure Chemical Vapor Deposition Reactor for the Deposition of Thin Films	500
Example 14.6	Reaction Synthesis Via MINLP	508
	References	513
	Supplementary References	514
15	Optimization in Large-Scale Plant Design and Operations	515
15.1	Process Simulators and Optimization Codes	518
15.2	Optimization Using Equation-Based Process Simulators	525
15.3	Optimization Using Modular-Based Simulators	537
	<i>15.3.1 Sequential Modular Methods / 15.3.2 Simultaneous Modular Methods / 15.3.3 Calculation of Derivatives</i>	
15.4	Summary	546
	References	546
	Supplementary References	548
16	Integrated Planning, Scheduling, and Control in the Process Industries	549
16.1	Plant Optimization Hierarchy	550
16.2	Planning and Scheduling	553
	<i>16.2.1 Planning / 16.2.2 Scheduling</i>	

16.3	Plantwide Management and Optimization	565
16.4	Unit Management and Control	567
	<i>16.4.1 Formulating the MPC Optimization Problem</i>	
16.5	Process Monitoring and Analysis	575
	References	579
	Supplementary References	581
Appendixes		583
A	Mathematical Summary	583
	<i>A.1 Definitions / A.2 Basic Matrix Operations / A.3 Linear Independence and Row Operations / A.4 Solution of Linear Equations / A.5 Eigenvalues, Eigenvectors / References / Supplementary References / Problems</i>	
B	Cost Estimation	603
	<i>B.1 Capital Costs / B.2 Operating Costs / B.3 Taking Account of Inflation / B.4 Predicting Revenues in an Economic-Based Objective Function / B.5 Project Evaluation / References</i>	
	Nomenclature	631
	Name Index	637
	Subject Index	643

PREFACE

THE CHEMICAL INDUSTRY has undergone significant changes during the past 25 years due to the increased cost of energy, increasingly stringent environmental regulations, and global competition in product pricing and quality. One of the most important engineering tools for addressing these issues is optimization. Modifications in plant design and operating procedures have been implemented to reduce costs and meet constraints, with an emphasis on improving efficiency and increasing profitability. Optimal operating conditions can be implemented via increased automation at the process, plant, and company levels, often called computer-integrated manufacturing, or CIM. As the power of computers has increased, following Moore's Law of doubling computer speeds every 18 months, the size and complexity of problems that can be solved by optimization techniques have correspondingly expanded. Effective optimization techniques are now available in software for personal computers—a capability that did not exist 10 years ago.

To apply optimization effectively in the chemical industries, both the theory and practice of optimization must be understood, both of which we explain in this book. We focus on those techniques and discuss software that offers the most potential for success and gives reliable results.

The book introduces the necessary tools for problem solving. We emphasize how to formulate optimization problems appropriately because many engineers and scientists find this phase of their decision-making process the most exasperating and difficult. The nature of the model often predetermines the optimization algorithm to be used. Because of improvements in optimization algorithms and software, the modeling step usually offers more challenges and choices than the selection of the optimization technique. Appropriate meshing of the optimization technique and the model are essential for success in optimization. In this book we omit rigorous optimization proofs, replacing them with geometric or plausibility arguments without sacrificing correctness. Ample references are cited for those who wish to explore the theoretical concepts in more detail.

The book contains three main sections. Part I describes how to specify the three key components of an optimization problem, namely the

1. Objective function
2. Process model
3. Constraints

Part I comprises three chapters that motivate the study of optimization by giving examples of different types of problems that may be encountered in chemical engineering. After discussing the three components in the previous list, we describe six steps that must be used in solving an optimization problem. A potential user of optimization must be able to translate a verbal description of the problem into the appropriate mathematical description. He or she should also understand how the problem formulation influences its solvability. We show how problem simplification, sensitivity analysis, and estimating the unknown parameters in models are important steps in model building. Chapter 3 discusses how the objective function should be developed. We focus on economic factors in this chapter and present several alternative methods of evaluating profitability.

Part II covers the theoretical and computational basis for proven techniques in optimization. The choice of a specific technique must mesh with the three components in the list. Part II begins with Chapter 4, which provides the essential conceptual background for optimization, namely the concepts of local and global optima, convexity, and necessary and sufficient conditions for an optimum. Chapter 5 follows with a brief explanation of the most commonly used one-dimensional search methods. Chapter 6 presents reliable unconstrained optimization and methods. Chapter 7 treats linear programming theory, applications, and software, using matrix methods. Chapter 8 covers recent advances in nonlinear programming methods and software, and Chapter 9 deals with optimization of discrete processes, highlighting mixed-integer programming problems and methods. We conclude Part II with a new chapter (for the second edition) on global optimization methods, such as tabu search, simulated annealing, and genetic algorithms. Only deterministic optimization problems are treated throughout the book because lack of space precludes discussing stochastic variables, constraints, and coefficients.

Although we include many simple applications in Parts I and II to illustrate the optimization techniques and algorithms, Part III of the book is exclusively devoted to illustrations and examples of optimization procedures, classified according to their applications: heat transfer and energy conservation (Chapter 11), separations (Chapter 12), fluid flow (Chapter 13), reactor design (Chapter 14), and plant design (Chapter 15), and a new chapter for the second edition on planning, scheduling, and control using optimization techniques (Chapter 16). Many students and professionals learn by example or analogy and often discover how to solve a problem by examining the solution to similar problems. By organizing applications of optimization in this manner, you can focus on a single class of applications of particular interest without having to review the entire book. We present a spectrum of modeling and solution methods in each of these chapters. The introduction to Part III lists each application classified by the technique employed. In some cases the

optimization method may be an analytical solution, leading to simple design rules; most examples illustrate numerical methods. In some applications the problem statement may be so complex that it cannot be explicitly written out, as in plant design and thus requires the use of a process simulator. No exercises are included in Part III, but an instructor can (1) modify the variables, parameters, conditions, or constraints in an example, and (2) suggest a different solution technique to obtain exercises for solution by students.

An understanding of optimization techniques does not require complex mathematics. We require as background only basic tools from multivariable calculus and linear algebra to explain the theory and computational techniques and provide you with an understanding of how optimization techniques work (or, in some cases, fail to work).

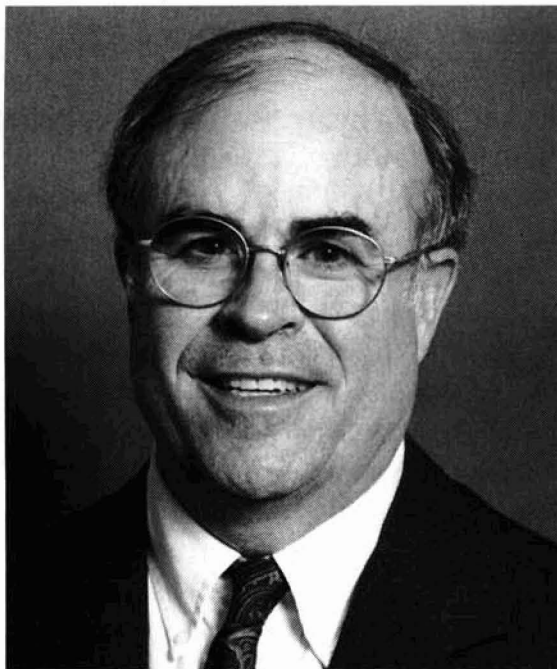
Presentation of each optimization technique is followed by examples to illustrate an application. We also have included many practically oriented homework problems. In university courses, this book could be used at the upper-division or the first-year graduate levels, either in a course focused on optimization or on process design. The book contains more than enough material for a 15-week course on optimization. Because of its emphasis on applications and short case studies in Chapters 11–16, it may also serve as one of the supplementary texts in a senior unit operations or design course.

In addition to use as a textbook, the book is also suitable for use in individual study, industrial practice, industrial short courses, and other continuing education programs.

We wish to acknowledge the helpful suggestions of several colleagues in developing this book, especially Yaman Arkun, Georgia Institute of Technology; Lorenz T. Biegler, Carnegie-Mellon University; James R. Couper, University of Arkansas; James R. Fair, University of Texas-Austin; Christodoulos Floudas, Princeton University; Fred Glover, University of Colorado; Ignacio Grossmann, Carnegie-Mellon University; K. Jayaraman, Michigan State University; I. Lefkowitz, Case Western Reserve University; Tom McAvoy, University of Maryland; János Pintér, Pintér Consulting Services; Larry Ricker, University of Washington; and Mark Stadtherr, University of Notre Dame. Several of the examples in Chapters 11–16 were provided by friends in industry and in universities and are acknowledged there. We also recognize the help of many graduate students in developing solutions to the examples, especially Juergen Hahn and Tyler Soderstrom for this edition.

T. F. Edgar
D. M. Himmelblau
L. S. Lasdon

ABOUT THE AUTHORS



THOMAS F. EDGAR holds the Abell Chair in chemical engineering at the University of Texas at Austin. He earned a B. S. in chemical engineering from the University of Kansas and a Ph. D. from Princeton University. Before receiving his doctorate, he was employed by Continental Oil Company. His professional honors include selection as the 1980 winner of the AIChE Colburn Award, ASEE Meriam-Wiley and Chemical Engineering Division Awards, ISA Education Award, and AIChE Computing in Chemical Engineering Award. He is listed in *Who's Who in America*.

He has published over 200 papers in the fields of process control, optimization, and mathematical modeling of processes such as separations, combustion, and microelectronics processing. He is coauthor of *Process Dynamics and Control*, published by Wiley in 1989. Dr. Edgar was chairman of the CAST Division of AIChE in 1986, president of the CACHE Corporation from 1981 to 1984, and president of AIChE in 1997.



DAVID M. HIMMELBLAU is the Paul D. and Betty Robertson Meek and American Petrofina Foundation Centennial Professor Emeritus in Chemical Engineering at the University of Texas at Austin. He received a B. S. degree from Massachusetts Institute of Technology and M. S. and Ph. D. degrees from the University of Washington. He has taught at the University of Texas for over 40 years. Prior to that time he worked for several companies including International Harvester Co., Simpson Logging Co., and Excel Battery Co. Among his more than 200 publications are 11 books including a widely used introductory book in chemical engi-

neering; books on process analysis and simulation, statistics, decomposition, fault detection in chemical processes; and nonlinear programming. He is a fellow of the American Institute of Chemical Engineers and served AIChE in many capacities, including as director. He also has been a CACHE trustee for many years, serving as president and later executive officer. He received the AIChE Founders Award and the CAST Division Computers in Chemical Engineering Award. His current areas of research are fault detection, sensor validation, and interactive learning via computer-based educational materials.



LEON LASDON holds the David Bruton Jr. Centennial Chair in Business Decision Support Systems in the Management Science and Information Systems Department, College of Business Administration, at the University of Texas at Austin and has taught there since 1977. He received a B. S. E. E. degree from Syracuse University and an M. S. E. E. degree and a Ph. D. in systems engineering from Case Institute of Technology.

Dr. Lasdon has published an award-winning text on large-scale systems optimization, and more than 100 articles in journals such as *Management Science*, *Operations Research*, *Mathematical Programming*, and the *INFORMS*

Journal on Computing. His research interests include optimization algorithms and software, and applications of optimization and other OR/MS methodologies. He is a coauthor of the Microsoft Excel Solver, and his optimization software is used in many industries and universities worldwide. He is consulted widely on problems involving OR/MS applications.

PART I

PROBLEM FORMULATION

Formulating the problem is perhaps the most crucial step in optimization. Problem formulation requires identifying the essential elements of a conceptual or verbal statement of a given application and organizing them into a prescribed mathematical form, namely,

1. The objective function (economic criterion)
2. The process model (constraints)

The objective function represents such factors as profit, cost, energy, and yield in terms of the key variables of the process being analyzed. The process model and constraints describe the interrelationships of the key variables. It is important to learn a systematic approach for assembling the physical and empirical relations and data involved in an optimization problem, and Chapters 1, 2, and 3 cover the recommended procedures. Chapter 1 presents six steps for optimization that can serve as a general guide for problem solving in design and operations analysis. Numerous examples of problem formulation in chemical engineering are presented to illustrate the steps.

Chapter 2 summarizes the characteristics of process models and explains how to build one. Special attention is focused on developing mathematical models, particularly empirical ones, by fitting empirical data using least squares, which itself is an optimization procedure.

Chapter 3 treats the most common type of objective function, the cost or revenue function. Historically, the majority of optimization applications have involved trade-offs between capital costs and operating costs. The nature of the trade-off depends on a number of assumptions such as the desired rate of return on investment, service life, depreciation method, and so on. While an objective function based on net present value is preferred for the purposes of optimization, discounted cash flow based on spreadsheet analysis can be employed as well.

It is important to recognize that many possible mathematical problem formulations can result from an engineering analysis, depending on the assumptions

made and the desired accuracy of the model. To solve an optimization problem, the mathematical formulation of the model must mesh satisfactorily with the computational algorithm to be used. A certain amount of artistry, judgment, and experience is therefore required during the problem formulation phase of optimization.

**THE NATURE AND ORGANIZATION OF
OPTIMIZATION PROBLEMS**

1.1 What Optimization Is All About 4
1.2 Why Optimize? 4
1.3 Scope and Hierarchy of Optimization 5
1.4 Examples of Applications of Optimization 8
1.5 The Essential Features of Optimization Problems 14
1.6 General Procedure for Solving Optimization Problems 18
1.7 Obstacles to Optimization 26
 References 27
 Supplementary References 27
 Problems 28

OPTIMIZATION IS THE use of specific methods to determine the most cost-effective and efficient solution to a problem or design for a process. This technique is one of the major quantitative tools in industrial decision making. A wide variety of problems in the design, construction, operation, and analysis of chemical plants (as well as many other industrial processes) can be resolved by optimization. In this chapter we examine the basic characteristics of optimization problems and their solution techniques and describe some typical benefits and applications in the chemical and petroleum industries.

1.1 WHAT OPTIMIZATION IS ALL ABOUT

A well-known approach to the principle of optimization was first scribbled centuries ago on the walls of an ancient Roman bathhouse in connection with a choice between two aspirants for emperor of Rome. It read—“De duobus malis, minus est semper aligendum”—of two evils, always choose the lesser.

Optimization pervades the fields of science, engineering, and business. In physics, many different optimal principles have been enunciated, describing natural phenomena in the fields of optics and classical mechanics. The field of statistics treats various principles termed “maximum likelihood,” “minimum loss,” and “least squares,” and business makes use of “maximum profit,” “minimum cost,” “maximum use of resources,” “minimum effort,” in its efforts to increase profits. A typical engineering problem can be posed as follows: A process can be represented by some equations or perhaps solely by experimental data. You have a single performance criterion in mind such as minimum cost. The goal of optimization is to find the values of the variables in the process that yield the best value of the performance criterion. A trade-off usually exists between capital and operating costs. The described factors—process or model and the performance criterion—constitute the optimization “problem.”

Typical problems in chemical engineering process design or plant operation have many (possibly an infinite number) solutions. Optimization is concerned with selecting the best among the entire set by efficient quantitative methods. Computers and associated software make the necessary computations feasible and cost-effective. To obtain useful information using computers, however, requires (1) critical analysis of the process or design, (2) insight about what the appropriate performance objectives are (i.e., what is to be accomplished), and (3) use of past experience, sometimes called engineering judgment.

1.2 WHY OPTIMIZE?

Why are engineers interested in optimization? What benefits result from using this method rather than making decisions intuitively? Engineers work to improve the initial design of equipment and strive to enhance the operation of that equipment once it is installed so as to realize the largest production, the greatest profit, the

minimum cost, the least energy usage, and so on. Monetary value provides a convenient measure of different but otherwise incompatible objectives, but not all problems have to be considered in a monetary (cost versus revenue) framework.

In plant operations, benefits arise from improved plant performance, such as improved yields of valuable products (or reduced yields of contaminants), reduced energy consumption, higher processing rates, and longer times between shutdowns. Optimization can also lead to reduced maintenance costs, less equipment wear, and better staff utilization. In addition, intangible benefits arise from the interactions among plant operators, engineers, and management. It is extremely helpful to systematically identify the objective, constraints, and degrees of freedom in a process or a plant, leading to such benefits as improved quality of design, faster and more reliable troubleshooting, and faster decision making.

Predicting benefits must be done with care. Design and operating variables in most plants are always coupled in some way. If the fuel bill for a distillation column is \$3000 per day, a 5-percent savings may justify an energy conservation project. In a unit operation such as distillation, however, it is incorrect to simply sum the heat exchanger duties and claim a percentage reduction in total heat required. A reduction in the reboiler heat duty may influence both the product purity, which can translate to a change in profits, and the condenser cooling requirements. Hence, it may be misleading to ignore the indirect and coupled effects that process variables have on costs.

What about the argument that the formal application of optimization is really not warranted because of the uncertainty that exists in the mathematical representation of the process or the data used in the model of the process? Certainly such an argument has some merit. Engineers have to use judgment in applying optimization techniques to problems that have considerable uncertainty associated with them, both from the standpoint of accuracy and the fact that the plant operating parameters and environs are not always static. In some cases it may be possible to carry out an analysis via deterministic optimization and then add on stochastic features to the analysis to yield quantitative predictions of the degree of uncertainty. Whenever the model of a process is idealized and the input and parameter data only known approximately, the optimization results must be treated judiciously. They can provide upper limits on expectations. Another way to evaluate the influence of uncertain parameters in optimal design is to perform a sensitivity analysis. It is possible that the optimum value of a process variable is unaffected by certain parameters (low sensitivity); therefore, having precise values for these parameters will not be crucial to finding the true optimum. We discuss how a sensitivity analysis is performed later on in this chapter.

1.3 SCOPE AND HIERARCHY OF OPTIMIZATION

Optimization can take place at many levels in a company, ranging from a complex combination of plants and distribution facilities down through individual plants, combinations of units, individual pieces of equipment, subsystems in a piece of

equipment, or even smaller entities (Beveridge and Schechter, 1970). Optimization problems can be found at all these levels. Thus, the scope of an optimization problem can be the entire company, a plant, a process, a single unit operation, a single piece of equipment in that operation, or any intermediate system between these. The complexity of analysis may involve only gross features or may examine minute detail, depending upon the use to which the results will be put, the availability of accurate data, and the time available in which to carry out the optimization. In a typical industrial company optimization can be used in three areas (levels): (1) management, (2) process design and equipment specification, and (3) plant operations (see Fig. 1.1).

Management makes decisions concerning project evaluation, product selection, corporate budget, investment in sales versus research and development, and new plant construction (i.e., when and where should new plants be constructed). At this level much of the available information may be qualitative or has a high degree of uncertainty. Many management decisions for optimizing some feature(s) of a large company therefore have the potential to be significantly in error when put into practice, especially if the timing is wrong. In general, the magnitude of the objective function, as measured in dollars, is much larger at the management level than at the other two levels.

Individuals engaged in process design and equipment specification are concerned with the choice of a process and nominal operating conditions. They answer questions such as: Do we design a batch process or a continuous process? How many reactors do we use in producing a petrochemical? What should the configurations of the plant be, and how do we arrange the processes so that the operating efficiency of the plant is at a maximum? What is the optimum size of a unit or combination of units? Such questions can be resolved with the aid of so-called process

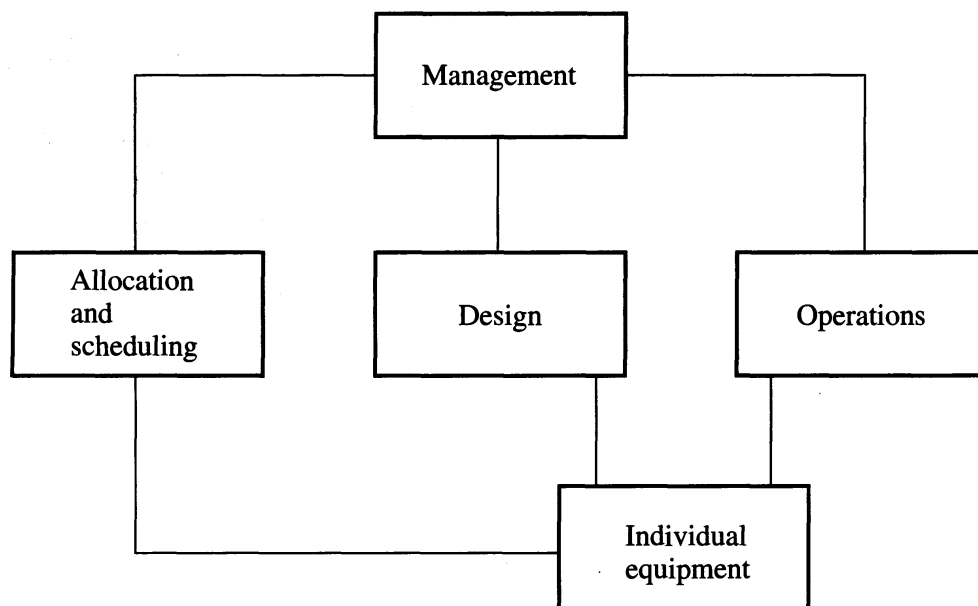


FIGURE 1.1
Hierarchy of levels of optimization.

design simulators or flowsheeting programs. These large computer programs carry out the material and energy balances for individual pieces of equipment and combine them into an overall production unit. Iterative use of such a simulator is often necessary to arrive at a desirable process flowsheet.

Other, more specific decisions are made in process design, including the actual choice of equipment (e.g., more than ten different types of heat exchangers are available) and the selection of construction materials of various process units.

The third constituency employing optimization operates on a totally different time scale than the other two. Process design and equipment specification is usually performed prior to the implementation of the process, and management decisions to implement designs are usually made far in advance of the process design step. On the other hand, optimization of operating conditions is carried out monthly, weekly, daily, hourly, or even, at the extreme, every minute. Plant operations are concerned with operating controls for a given unit at certain temperatures, pressures, or flowrates that are the best in some sense. For example, the selection of the percentage of excess air in a process heater is critical and involves balancing the fuel-air ratio to ensure complete combustion while making the maximum use of the heating potential of the fuel.

Plant operations deal with the allocation of raw materials on a daily or weekly basis. One classical optimization problem, which is discussed later in this text, is the allocation of raw materials in a refinery. Typical day-to-day optimization in a plant minimizes steam consumption or cooling water consumption.

Plant operations are also concerned with the overall picture of shipping, transportation, and distribution of products to engender minimal costs. For example, the frequency of ordering, the method of scheduling production, and scheduling delivery are critical to maintaining a low-cost operation.

The following attributes of processes affecting costs or profits make them attractive for the application of optimization:

1. *Sales limited by production*: If additional products can be sold beyond current capacity, then economic justification of design modifications is relatively easy. Often, increased production can be attained with only slight changes in operating costs (raw materials, utilities, etc.) and with no change in investment costs. This situation implies a higher profit margin on the incremental sales.
2. *Sales limited by market*: This situation is susceptible to optimization only if improvements in efficiency or productivity can be obtained; hence, the economic incentive for implementation in this case may be less than in the first example because no additional products are made. Reductions in unit manufacturing costs (via optimizing usage of utilities and feedstocks) are generally the main targets.
3. *Large unit throughputs*: High production volume offers great potential for increased profits because small savings in production costs per unit are greatly magnified. Most large chemical and petroleum processes fall into this classification.
4. *High raw material or energy consumption*: Significant savings can be made by reducing consumption of those items with high unit costs.

5. *Product quality exceeds product specifications*: If the product quality is significantly better than that required by the customer, higher than necessary production costs and wasted capacity may occur. By operating close to customer specification (constraints), cost savings can be obtained.
6. *Losses of valuable components through waste streams*: The chemical analysis of various plant exit streams, both to the air and water, should indicate if valuable materials are being lost. Adjustment of air–fuel ratios in furnaces to minimize hydrocarbon emissions and hence fuel consumption is one such example. Pollution regulations also influence permissible air and water emissions.
7. *High labor costs*: In processes in which excessive handling is required, such as in batch operation, bulk quantities can often be handled at lower cost and with a smaller workforce. Revised layouts of facilities can reduce costs. Sometimes no direct reduction in the labor force results, but the intangible benefits of a lessened workload can allow the operator to assume greater responsibility.

Two valuable sources of data for identifying opportunities for optimization include (1) profit and loss statements for the plant or the unit and (2) the periodic operating records for the plant. The profit and loss statement contains much valuable information on sales, prices, manufacturing costs, and profits, and the operating records present information on material and energy balances, unit efficiencies, production levels, and feedstock usage.

Because of the complexity of chemical plants, complete optimization of a given plant can be an extensive undertaking. In the absence of complete optimization we often rely on “incomplete optimization,” a special variety of which is termed *suboptimization*. Suboptimization involves optimization for one phase of an operation or a problem while ignoring some factors that have an effect, either obvious or indirect, on other systems or processes in the plant. Suboptimization is often necessary because of economic and practical considerations, limitations on time or personnel, and the difficulty of obtaining answers in a hurry. Suboptimization is useful when neither the problem formulation nor the available techniques permits obtaining a reasonable solution to the full problem. In most practical cases, suboptimization at least provides a rational technique for approaching an optimum.

Recognize, however, that suboptimization of all elements does *not* necessarily ensure attainment of an overall optimum for the *entire* system. Subsystem objectives may not be compatible nor mesh with overall objectives.

1.4 EXAMPLES OF APPLICATIONS OF OPTIMIZATION

Optimization can be applied in numerous ways to chemical processes and plants. Typical projects in which optimization has been used include

1. Determining the best sites for plant location.
2. Routing tankers for the distribution of crude and refined products.
3. Sizing and layout of a pipeline.
4. Designing equipment and an entire plant.

5. Scheduling maintenance and equipment replacement.
6. Operating equipment, such as tubular reactors, columns, and absorbers.
7. Evaluating plant data to construct a model of a process.
8. Minimizing inventory charges.
9. Allocating resources or services among several processes.
10. Planning and scheduling construction.

These examples provide an introduction to the types of variables, objective functions, and constraints that will be encountered in subsequent chapters.

In this section we provide four illustrations of “optimization in practice.” that is, optimization of process operations and design. These examples will help illustrate the general features of optimization problems, a topic treated in more detail in Section 1.5.

EXAMPLE 1.1 OPTIMAL INSULATION THICKNESS

Insulation design is a classic example of overall cost saving that is especially pertinent when fuel costs are high. The addition of insulation should save money through reduced heat losses; on the other hand, the insulation material can be expensive. The amount of added insulation needed can be determined by optimization.

Assume that the bare surface of a vessel is at 700°F with an ambient temperature of 70°F. The surface heat loss is 4000 Btu/(h)(ft²). Add 1 in. of calcium silicate insulation and the loss will drop to 250 Btu/(h)(ft²). At an installed cost of \$4.00/ft² and a cost of energy at \$5.00/10⁶ Btu, a savings of \$164 per year (8760 hours of operation) per square foot would be realized. A simplified payback calculation shows a payback period of

$$\frac{\$4.00/(\text{ft}^2)}{\$164/(\text{ft}^2)(\text{year})} = 0.0244 \text{ year, or 9 days}$$

As additional inches of insulation are added, the increments must be justified by the savings obtained. Figure E1.1 shows the outcome of adding more layers of insulation. Since insulation can only be added in 0.5-in. increments, the possible capital costs are shown as a series of dots; these costs are prorated because the insulation lasts for several years before having to be replaced. In Figure E1.1 the energy loss cost is a continuous curve because it can be calculated directly from heat transfer principles. The total cost is also shown as a continuous function. Note that at some point total costs begin increasing as the insulation thickness increases because little or no benefit in heat conservation results. The trade-off between energy cost and capital cost, and the optimum insulation thickness, can be determined by optimization. Further discussion of capital versus operating costs appears in Chapter 3; in particular, see Example 3.3.

EXAMPLE 1.2 OPTIMAL OPERATING CONDITIONS OF A BOILER

Another example of optimization can be encountered in the operation of a boiler. Engineers focus attention on utilities and powerhouse operations within refineries and

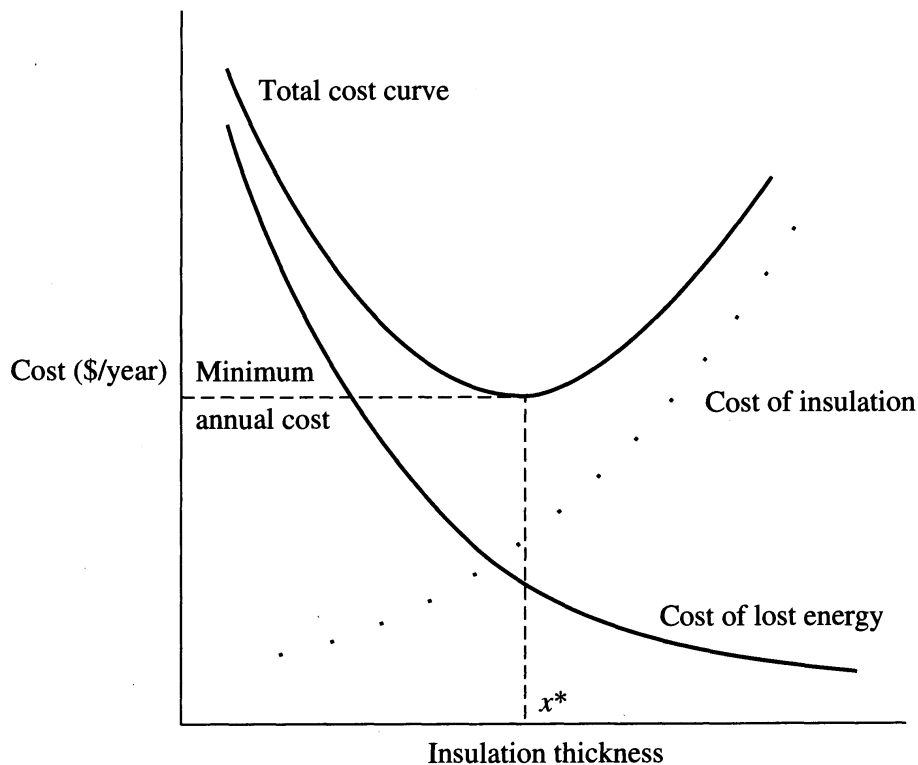


FIGURE E1.1

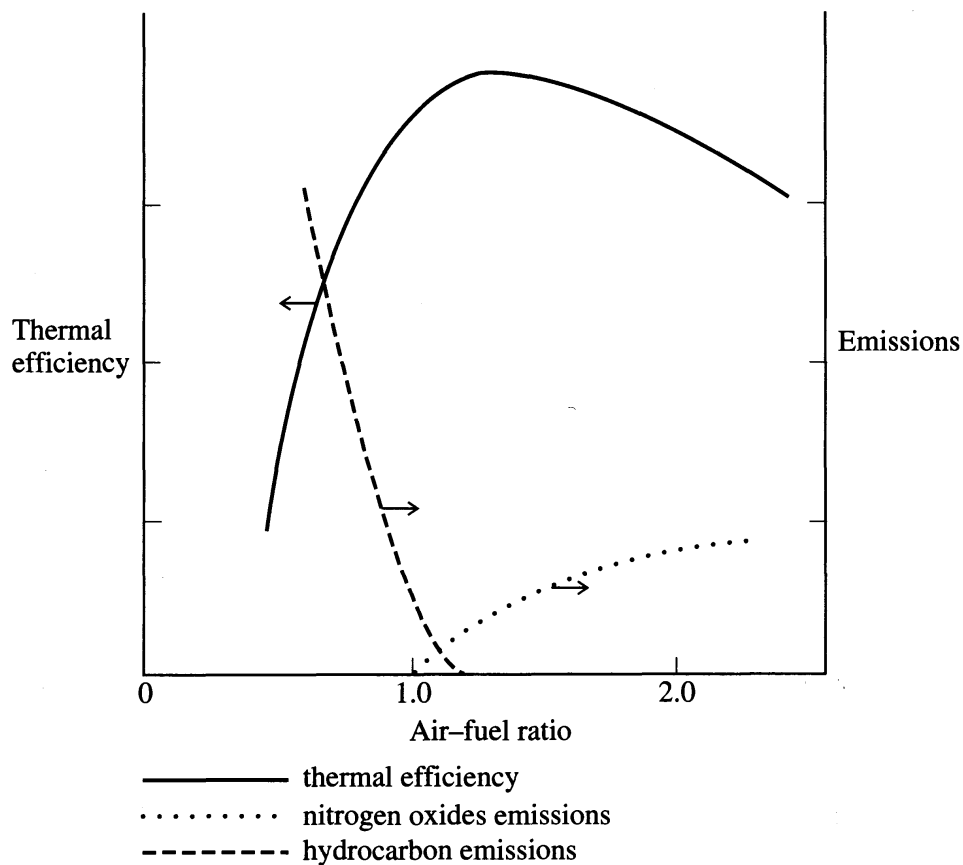
The effect of insulation thickness on total cost (x^* = optimum thickness). Insulation can be purchased in 0.5-in. increments. (The total cost function is shown as a smooth curve for convenience, although the sum of the two costs would not actually be smooth.)

process plants because of the large amounts of energy consumed by these plants and the potential for significant reduction in the energy required for utilities generation and distribution. Control of environmental emissions adds complexity and constraints in optimizing boiler operations. In a boiler it is desirable to optimize the air–fuel ratio so that the thermal efficiency is maximized; however, environmental regulations encourage operation under fuel-rich conditions and lower combustion temperatures in order to reduce the emissions of nitrogen oxides (NO_x). Unfortunately, such operating conditions also decrease efficiency because some unburned fuel escapes through the stacks, resulting in an increase in undesirable hydrocarbon (HC) emissions. Thus, a conflict in operating criteria arises.

Figure E1.2a illustrates the trade-offs between efficiency and emissions, suggesting that more than one performance criterion may exist: We are forced to consider maximizing efficiency versus minimizing emissions, resulting in some compromise of the two objectives.

Another feature of boiler operations is the widely varying demands caused by changes in process operations, plant unit start-ups and shutdowns, and daily and seasonal cycles. Because utility equipment is often operated in parallel, demand swings commonly affect when another boiler, turbine, or other piece of equipment should be brought on line and which one it should be.

Determining this is complicated by the feature that most powerhouse equipment cannot be operated continuously all the way down to the idle state, as illustrated by Figure E1.2b for boilers and turbines. Instead, a range of continuous operation may

**FIGURE E1.2a**

Efficiency and emissions of a boiler as a function of air-fuel ratio. (1.0 = stoichiometric air-fuel ratio.)

exist for certain conditions, but a discrete jump to a different set of conditions (here idling conditions) may be required if demand changes. In formulating many optimization problems, discrete variables (on-off, high-low, integer 1, 2, 3, 4, etc.) must be accommodated.

EXAMPLE 1.3 OPTIMUM DISTILLATION REFLUX

Prior to 1974, when fuel costs were low, distillation column trains used a strategy involving the substantial consumption of utilities such as steam and cooling water in order to maximize separation (i.e., product purity) for a given tower. However, the operation of any one tower involves certain limitations or constraints on the process, such as the condenser duty, tower tray flooding, or reboiler duty.

The need for energy conservation suggests a different objective, namely minimizing the reflux ratio. In this circumstance, one can ask: How low can the reflux ratio be set? From the viewpoint of optimization, there is an economic minimum value below which the energy savings are less than the cost of product quality degradation. Figures E1.3a and E1.3b illustrate both alternatives. Operators tend to over-reflux a column because this strategy makes it easier to stay well within the product

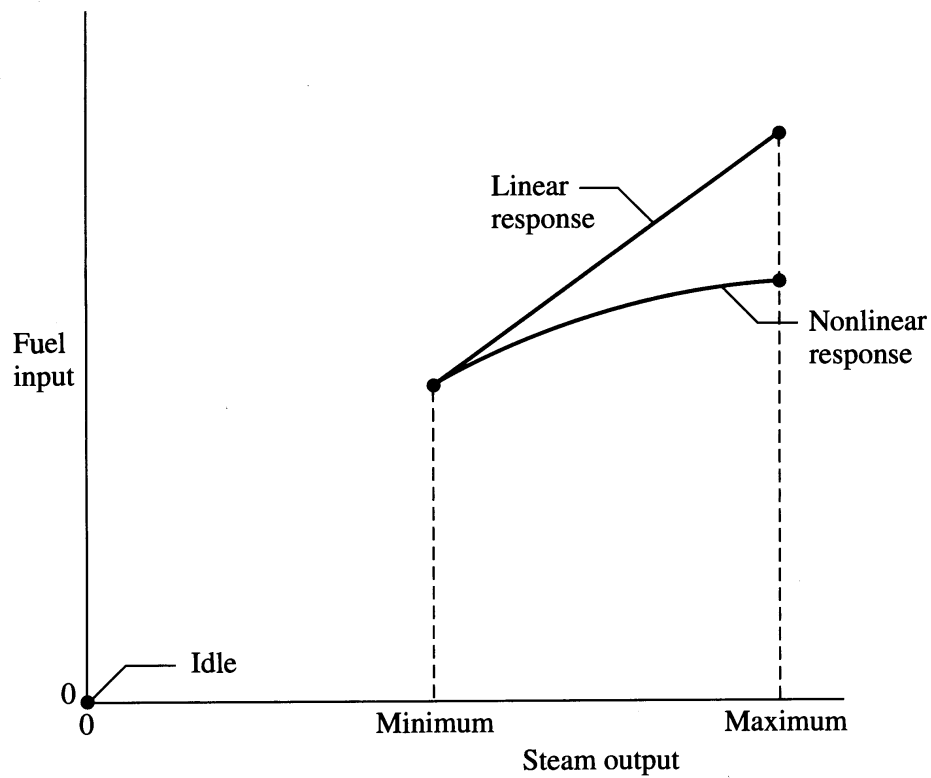


FIGURE E1.2b
Discontinuity in operating regimen.

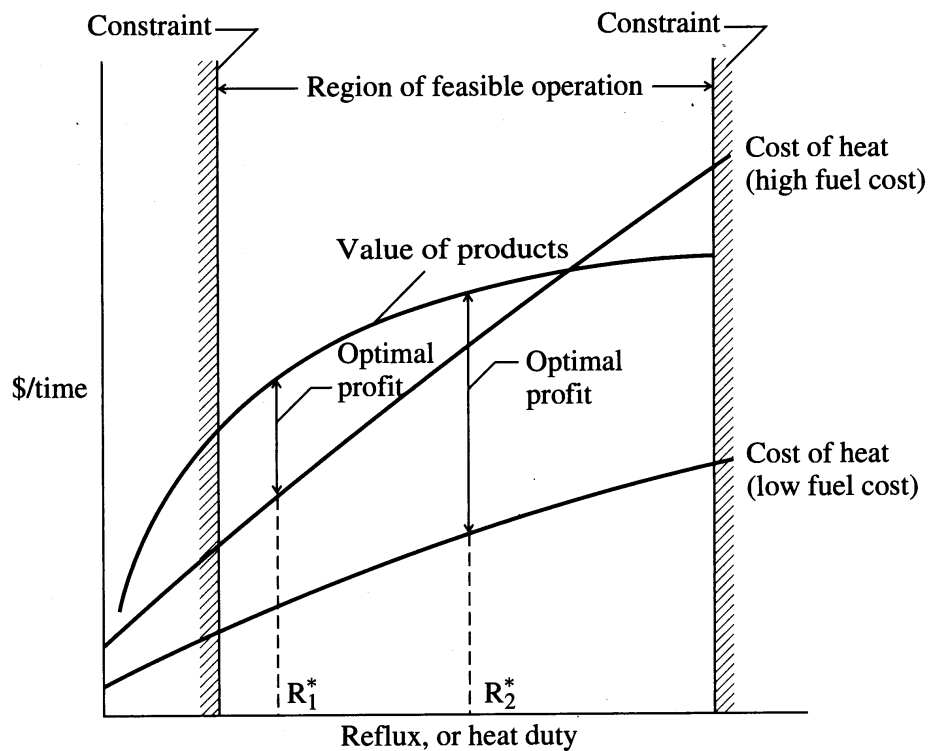


FIGURE E1.3a
Illustration of optimal reflux for different fuel costs.

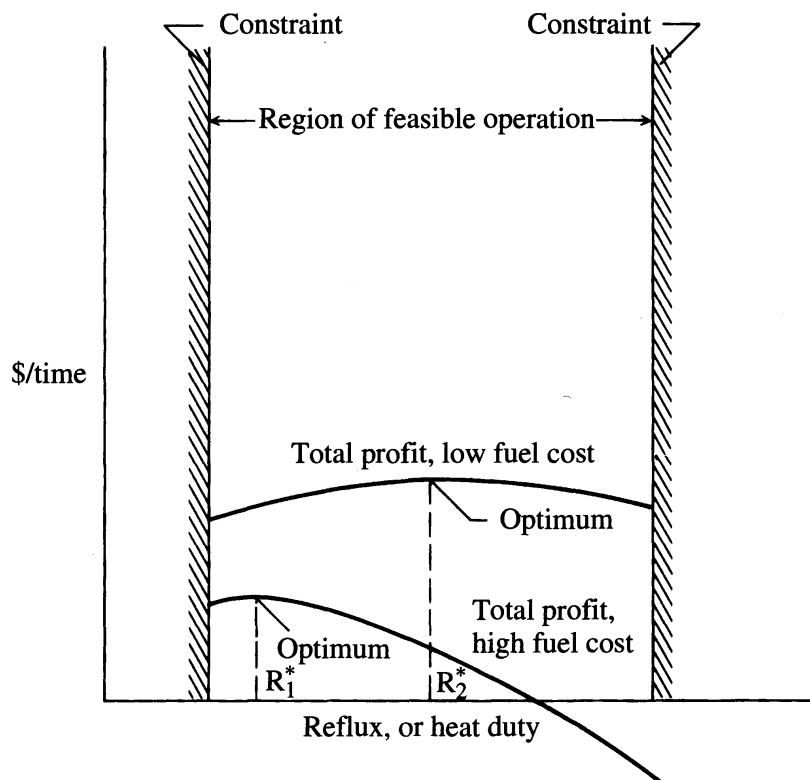


FIGURE E1.3b
Total profit for different fuel costs.

specifications. Often columns are operated with a fixed flow control for reflux so that the reflux ratio is higher than needed when feed rates drop off. This issue is discussed in more detail in Chapter 12.

EXAMPLE 1.4 MULTIPLANT PRODUCT DISTRIBUTION

A common problem encountered in large chemical companies involves the distribution of a single product (Y) manufactured at several plant locations. Generally, the product needs to be delivered to several customers located at various distances from each plant. It is, therefore, desirable to determine how much Y must be produced at each of m plants (Y_1, Y_2, \dots, Y_m) and how, for example, Y_m should be allocated to each of n demand points ($Y_{m1}, Y_{m2}, \dots, Y_{mn}$). The cost-minimizing solution to this problem not only involves the transportation costs between each supply and demand point but also the production cost versus capacity curves for each plant. The individual plants probably vary with respect to their nominal production rate, and some plants may be more efficient than others, having been constructed at a later date. Both of these factors contribute to a unique functionality between production cost and production rate. Because of the particular distribution of transportation costs, it may be

desirable to manufacture more product from an old, inefficient plant (at higher cost) than from a new, efficient one because new customers may be located very close to the old plant. On the other hand, if the old plant is operated far above its design rate, costs could become exorbitant, forcing a reallocation by other plants in spite of high transportation costs. In addition, no doubt constraints exist on production levels from each plant that also affect the product distribution plan.

1.5 THE ESSENTIAL FEATURES OF OPTIMIZATION PROBLEMS

Because the solution of optimization problems involves various features of mathematics, the formulation of an optimization problem must use mathematical expressions. Such expressions do not necessarily need to be very complex. Not all problems can be stated or analyzed quantitatively, but we will restrict our coverage to quantitative methods. From a practical viewpoint, it is important to mesh properly the problem statement with the anticipated solution technique.

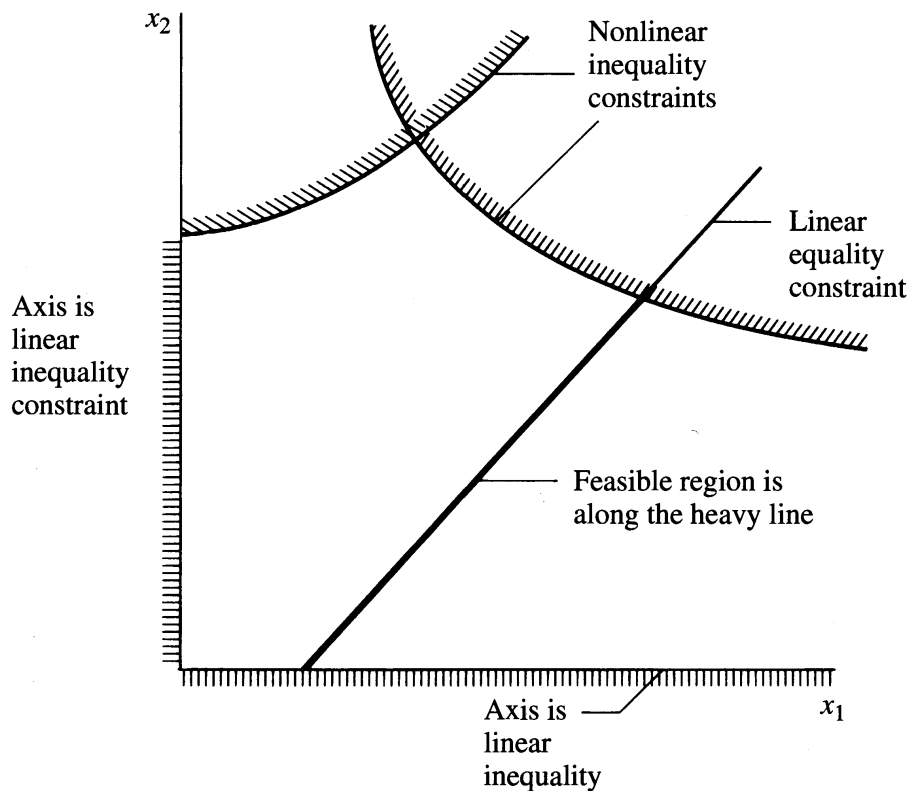
A wide variety of optimization problems have amazingly similar structures. Indeed, it is this similarity that has enabled the recent progress in optimization techniques. Chemical engineers, petroleum engineers, physicists, chemists, and traffic engineers, among others, have a common interest in precisely the same mathematical problem structures, each with a different application in the real world. We can make use of this structural similarity to develop a framework or methodology within which any problem can be studied. This section describes how any process problem, complex or simple, for which one desires the optimal solution should be organized. To do so, you must (a) consider the model representing the process and (b) choose a suitable objective criterion to guide the decision making.

Every optimization problem contains three essential categories:

1. At least one objective function to be optimized (profit function, cost function, etc.).
2. Equality constraints (equations).
3. Inequality constraints (inequalities).

Categories 2 and 3 constitute the model of the process or equipment; category 1 is sometimes called the *economic model*.

By a *feasible solution* of the optimization problem we mean a set of variables that satisfy categories 2 and 3 to the desired degree of precision. Figure 1.2 illustrates the feasible region or the region of feasible solutions defined by categories 2 and 3. In this case the feasible region consists of a line bounded by two inequality constraints. An *optimal solution* is a set of values of the variables that satisfy the components of categories 2 and 3; this solution also provides an optimal value for the function in category 1. In most cases the optimal solution is a unique one; in some it is not. If you formulate the optimization problem so that there are no residual degrees of freedom among the variables in categories 2 and 3, optimization is

**FIGURE 1.2**

Feasible region for an optimization problem involving two independent variables. The dashed lines represent the side of the inequality constraints in the plane that form part of the infeasible region. The heavy line shows the feasible region.

not needed to obtain a solution for a problem. More specifically, if m_e equals the number of independent consistent equality constraints and m_i equals the number of independent inequality constraints that are satisfied as equalities (equal to zero), and if the number of variables whose values are unknown is equal to $m_e + m_i$, then at least one solution exists for the relations in components 2 and 3 regardless of the optimization criterion. (Multiple solutions may exist when models in categories 2 and 3 are composed of nonlinear relations.) If a unique solution exists, no optimization is needed to obtain a solution—one just solves a set of equations and need not worry about optimization methods because the unique feasible solution is by definition the optimal one.

On the other hand, if more process variables whose values are unknown exist in category 2 than there are independent equations, the process model is called *underdetermined*; that is, the model has an infinite number of feasible solutions so that the objective function in category 1 is the additional criterion used to reduce the number of solutions to just one (or a few) by specifying what is the “best” solution. Finally, if the equations in category 2 contain more independent equations

than variables whose values are unknown, the process model is *overdetermined* and no solution satisfies all the constraints exactly. To resolve the difficulty, we sometimes choose to relax some or all of the constraints. A typical example of an overdetermined model might be the reconciliation of process measurements for a material balance. One approach to yield the desired material balance would be to resolve the set of inconsistent equations by minimizing the sum of the errors of the set of equations (usually by a procedure termed *least squares*).

In this text the following notation will be used for each category of the optimization problem:

$$\text{Minimize: } f(\mathbf{x}) \quad \text{objective function} \quad (a)$$

$$\text{Subject to: } \mathbf{h}(\mathbf{x}) = \mathbf{0} \quad \text{equality constraints} \quad (b)$$

$$\mathbf{g}(\mathbf{x}) \geq \mathbf{0} \quad \text{inequality constraints} \quad (c)$$

where \mathbf{x} is a vector of n variables (x_1, x_2, \dots, x_n), $\mathbf{h}(\mathbf{x})$ is a vector of equations of dimension m_1 , and $\mathbf{g}(\mathbf{x})$ is a vector of inequalities of dimension m_2 . The total number of constraints is $m = (m_1 + m_2)$.

EXAMPLE 1.5 OPTIMAL SCHEDULING: FORMULATION OF THE OPTIMIZATION PROBLEM

In this example we illustrate the formulation of the components of an optimization problem.

We want to schedule the production in two plants, *A* and *B*, each of which can manufacture two products: 1 and 2. How should the scheduling take place to maximize profits while meeting the market requirements based on the following data:

Plant	Material processed (lb/day)		Profit (\$/lb)	
	1	2	1	2
<i>A</i>	M_{A1}	M_{A2}	S_{A1}	S_{A2}
<i>B</i>	M_{B1}	M_{B2}	S_{B1}	S_{B2}

How many days per year (365 days) should each plant operate processing each kind of material? *Hints*: Does the table contain the variables to be optimized? How do you use the information mathematically to formulate the optimization problem? What other factors must you consider?

Solution. How should we start to convert the words of the problem into mathematical statements? First, let us define the variables. There will be four of them (t_{A1}, t_{A2}, t_{B1} , and t_{B2} , designated as a set by the vector \mathbf{t}) representing, respectively, the number of days per year each plant operates on each material as indicated by the subscripts.

What is the objective function? We select the annual profit so that

$$f(\mathbf{t}) = t_{A1}M_{A1}S_{A1} + t_{A2}M_{A2}S_{A2} + t_{B1}M_{B1}S_{B1} + t_{B2}M_{B2}S_{B2} \quad (a)$$

Next, do any equality constraints evolve from the problem statement or from implicit assumptions? If each plant runs 365 days per year, two equality constraints arise:

$$t_{A1} + t_{A2} = 365 \quad (b)$$

$$t_{B1} + t_{B2} = 365 \quad (c)$$

Finally, do any inequality constraints evolve from the problem statement or implicit assumptions? On first glance it may appear that there are none, but further thought indicates t must be nonnegative since negative values of t have no physical meaning:

$$t_{Ai} \geq 0 \quad i = 1, 2 \quad (d)$$

$$t_{Bi} \geq 0 \quad i = 1, 2 \quad (e)$$

Do negative values of the coefficients S have physical meaning?

Other inequality constraints might be added after further analysis, such as a limitation on the total amount of material 2 that can be sold (L_1):

$$t_{A2}M_{A2} + t_{B2}M_{B2} \leq L_1 \quad (f)$$

or a limitation on production rate for each product at each plant, namely

$$M_{A1} \leq L_2$$

$$M_{A2} \leq L_3 \quad (g)$$

$$M_{B1} \leq L_4$$

$$M_{B2} \leq L_5$$

To find the optimal \mathbf{t} , we need to optimize (a) subject to constraints (b) to (g).

EXAMPLE 1.6 MATERIAL BALANCE RECONCILIATION

Suppose the flow rates entering and leaving a process are measured periodically. Determine the best value for stream A in kg/h for the process shown from the three hourly measurements indicated of B and C in Figure E1.6, assuming steady-state operation at a fixed operating point. The process model is

$$M_A + M_C = M_B \quad (a)$$

where M is the mass per unit time of throughput.

Solution. We need to set up the objective function first. Let us minimize the sum of the squares of the deviations between input and output as the criterion so that the objective function becomes

$$f(M_A) = (M_A + 11.1 - 92.4)^2 + (M_A + 10.8 - 94.3)^2 + (M_A + 11.4 - 93.8)^2 \quad (b)$$

A sum of squares is used since this guarantees that $f > 0$ for all values of M_A ; a minimum at $f = 0$ implies no error.

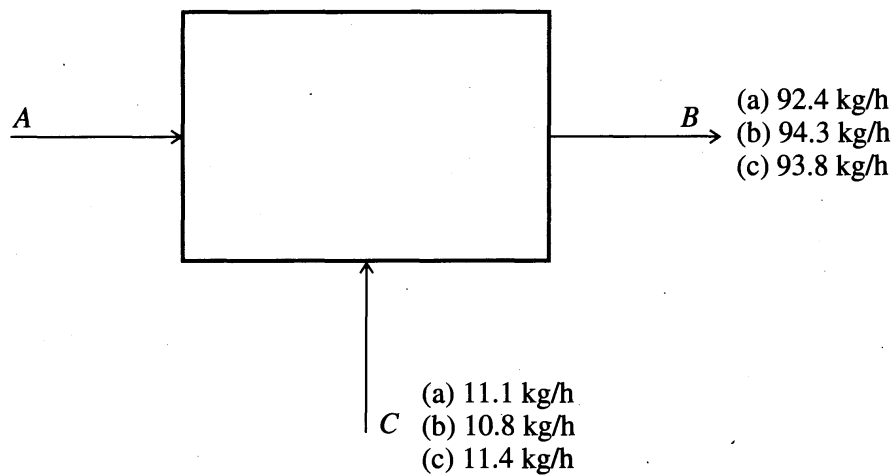


FIGURE E1.6

No equality constraints remain in the problem. Are there any inequality constraints? (*Hint*: What about M_A ?) The optimum value of M_A can be found by differentiating f with respect to M_A ; this leads to an optimum value for M_A of 82.4 and is the same result as that obtained by computing from the averaged measured values, $M_A = \bar{M}_B - \bar{M}_C$. Other methods of reconciling material (and energy) balances are discussed by Romagnoli and Sanchez (1999).

1.6 GENERAL PROCEDURE FOR SOLVING OPTIMIZATION PROBLEMS

No single method or algorithm of optimization can be applied efficiently to all problems. The method chosen for any particular case depends primarily on (1) the character of the objective function and whether it is known explicitly, (2) the nature of the constraints, and (3) the number of independent and dependent variables.

Table 1.1 lists the six general steps for the analysis and solution of optimization problems. You do not have to follow the cited order exactly, but you should cover all of the steps eventually. Shortcuts in the procedure are allowable, and the easy steps can be performed first. Each of the steps will be examined in more detail in subsequent chapters.

Remember, the general objective in optimization is to choose a set of values of the variables subject to the various constraints that produce the desired optimum response for the chosen objective function.

Steps 1, 2, and 3 deal with the mathematical definition of the problem, that is, identification of variables, specification of the objective function, and statement of the constraints. We devote considerable attention to problem formulation in the remainder of this chapter, as well as in Chapters 2 and 3. If the process to be optimized is very complex, it may be necessary to reformulate the problem so that it can be solved with reasonable effort.

Step 4 suggests that the mathematical statement of the problem be simplified as much as possible without losing the essence of the problem. First, you might

TABLE 1.1
The six steps used to solve optimization problems

-
1. Analyze the process itself so that the process variables and specific characteristics of interest are defined; that is, make a list of all of the variables.
 2. Determine the criterion for optimization, and specify the objective function in terms of the variables defined in step 1 together with coefficients. This step provides the performance model (sometimes called the economic model when appropriate).
 3. Using mathematical expressions, develop a valid process or equipment model that relates the input–output variables of the process and associated coefficients. Include both equality and inequality constraints. Use well-known physical principles (mass balances, energy balances), empirical relations, implicit concepts, and external restrictions. Identify the independent and dependent variables to get the number of degrees of freedom.
 4. If the problem formulation is too large in scope:
 - (a) break it up into manageable parts or
 - (b) simplify the objective function and model
 5. Apply a suitable optimization technique to the mathematical statement of the problem.
 6. Check the answers, and examine the sensitivity of the result to changes in the coefficients in the problem and the assumptions.
-

decide to ignore those variables that have an insignificant effect on the objective function. This step can be done either ad hoc, based on engineering judgment, or by performing a mathematical analysis and determining the weights that should be assigned to each variable via simulation. Second, a variable that appears in a simple form within an equation can be eliminated; that is, it can be solved for explicitly and then eliminated from other equations, the inequalities, and the objective function. Such variables are then deemed to be dependent variables.

As an example, in heat exchanger design, you might initially include the following variables in the problem: heat transfer surface, flow rates, number of shell passes, number of tube passes, number and spacing of the baffles, length of the exchanger, diameter of the tubes and shell, the approach temperature, and the pressure drop. Which of the variables are independent and which are not? This question can become quite complicated in a problem with many variables. You will find that each problem has to be analyzed and treated as an individual case; generalizations are difficult. Often the decision is quite arbitrary although instinct indicates that the controllable variables be initially selected as the independent ones.

If an engineer is familiar with a particular heat exchanger system, he or she might decide that certain variables can be ignored based on the notion of the controlling or dominant heat transfer coefficient. In such a case only one of the flowing streams is important in terms of calculating the heat transfer in the system, and the engineer might decide, at least initially, to eliminate from consideration those variables related to the other stream.

A third strategy can be carried out when the problem has many constraints and many variables. We assume that some variables are fixed and let the remainder of the variables represent degrees of freedom (independent variables) in the optimization procedure. For example, the optimum pressure of a distillation column might occur at the minimum pressure (as limited by condenser cooling).

Finally, analysis of the objective function may permit some simplification of the problem. For example, if one product (*A*) from a plant is worth \$30 per pound and all other products from the plant are worth \$5 or less per pound, then we might initially decide to maximize the production of *A* only.

Step 5 in Table 1.1 involves the computation of the optimum point. Quite a few techniques exist to obtain the optimal solution for a problem. We describe several methods in detail later on. In general, the solution of most optimization problems involves the use of a computer to obtain numerical answers. It is fair to state that over the past 20 years, substantial progress has been made in developing efficient and robust digital methods for optimization calculations. Much is known about which methods are most successful, although comparisons of candidate methods often are ad hoc, based on test cases of simple problems. Virtually all numerical optimization methods involve iteration, and the effectiveness of a given technique often depends on a good first guess as to the values of the variables at the optimal solution.

The last entry in Table 1.1 involves checking the candidate solution to determine that it is indeed optimal. In some problems you can check that the sufficient conditions for an optimum are satisfied. More often, an optimal solution may exist, yet you cannot demonstrate that the sufficient conditions are satisfied. All you can do is show by repetitive numerical calculations that the value of the objective function is superior to all known alternatives. A second consideration is the sensitivity of the optimum to changes in parameters in the problem statement. A sensitivity analysis for the objective function value is important and is illustrated as part of the next example.

EXAMPLE 1.7 THE SIX STEPS OF OPTIMIZATION FOR A MANUFACTURING PROBLEM

This example examines a simple problem in detail so that you can understand how to execute the steps for optimization listed in Table 1.1. You also will see in this example that optimization can give insight into the nature of optimal operations and how optimal results might compare with the simple or arbitrary rules of thumb so often used in practice.

Suppose you are a chemical distributor who wishes to optimize the inventory of a specialty chemical. You expect to sell Q barrels of this chemical over a given year at a fixed price with demand spread evenly over the year. If $Q = 100,000$ barrels (units) per year, you must decide on a production schedule. Unsold production is kept in inventory. To determine the optimal production schedule you must quantify those aspects of the problem that are important from a cost viewpoint [Baumol (1972)].

Step 1. One option is to produce 100,000 units in one run at the beginning of the year and allow the inventory to be reduced to zero at the end of the year (at which time

another 100,000 units are manufactured). Another option is to make ten runs of 10,000 apiece. It is clear that much more money is tied up in inventory with the former option than in the latter. Funds tied up in inventory are funds that could be invested in other areas or placed in a savings account. You might therefore conclude that it would be cheaper to make the product ten times a year.

However, if you extend this notion to an extreme and make 100,000 production runs of one unit each (actually one unit every 315 seconds), the decision obviously is impractical, since the cost of producing 100,000 units, one unit at a time, will be exorbitant. It therefore appears that the desired operating procedure lies somewhere in between the two extremes. To arrive at some quantitative answer to this problem, first define the three operating variables that appear to be important: number of units of each run (D), the number of runs per year (n), and the total number of units produced per year (Q). Then you must obtain details about the costs of operations. In so doing, a cost (objective) function and a mathematical model will be developed, as discussed later on. After obtaining a cost model, any constraints on the variables are identified, which allows selection of independent and dependent variables.

Step 2. Let the business costs be split up into two categories: (1) the carrying cost or the cost of inventory and (2) the cost of production. Let D be the number of units produced in one run, and let Q (annual production level) be assigned a known value. If the problem were posed so that a minimum level of inventory is specified, it would not change the structure of the problem.

The cost of the inventory not only includes the cost of the money tied up in the inventory, but also a storage cost, which is a function of the inventory size. Warehouse space must exist to store all the units produced in one run. In the objective function, let the cost of carrying the inventory be K_1D , where the parameter K_1 essentially lumps together the cost of working capital for the inventory itself and the storage costs.

Assume that the annual production cost in the objective function is proportional to the number of production runs required. The cost per run is assumed to be a linear function of D , given by the following equation:

$$\text{Cost per run} = K_2 + K_3D \quad (a)$$

The cost parameter K_2 is a setup cost and denotes a fixed cost of production—equipment must be made ready, cleaned, and so on. The parameter K_3 is an operating cost parameter. The operating cost is assumed to be proportional to the number of units manufactured. Equation (a) may be an unrealistic assumption because the incremental cost of manufacturing could decrease somewhat for large runs; consequently, instead of a linear function, you might choose a nonlinear cost function of the form

$$\text{Cost per run} = K_2 + K_4D^{1/2} \quad (b)$$

as is shown in Figure E1.7. The effect of this alternative assumption will be discussed later. The annual production cost can be found by multiplying either Equation (a) or (b) by the number n of production runs per year.

The total annual manufacturing cost C for the product is the sum of the carrying costs and the production costs, namely

$$C = K_1D + n(K_2 + K_3D) \quad (c)$$

Step 3. The objective function in (c) is a function of two variables: D and n . However, D and n are directly related, namely $n = Q/D$. Therefore, only one independent

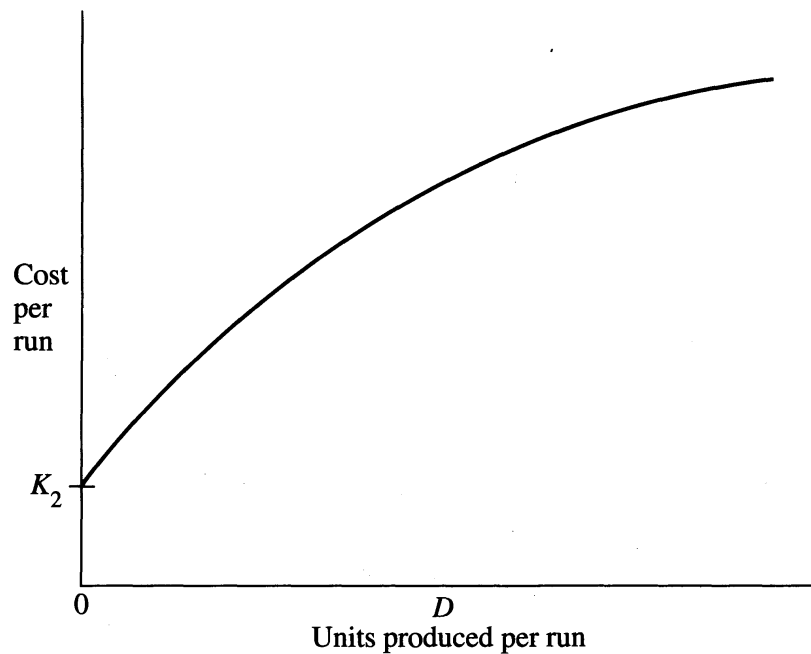


FIGURE E1.7
Nonlinear cost function for manufacturing.

variable exists for this problem, which we select to be D . The dependent variable is therefore n . Eliminating n from the objective function in (c) gives

$$C = K_1D + \frac{K_2Q}{D} + K_3Q \quad (d)$$

What other constraints exist in this problem? None are stated explicitly, but several implicit constraints exist. One of the assumptions made in arriving at Equation (c) is that over the course of one year, production runs of integer quantities may be involved. Can D be treated as a continuous variable? Such a question is crucial prior to using differential calculus to solve the problem. The occurrence of integer variables in principle prevents the direct calculation of derivatives of functions of integer variables. In the simple example here, with D being the only variable and a large one, you can treat D as continuous. After obtaining the optimal D , the practical value for D is obtained by rounding up or down. There is no guarantee that $n = Q/D$ is an integer; however, as long as you operate from year to year there should be no restriction on n .

What other constraints exist? You know that D must be positive. Do any equality constraints relate D to the other known parameters of the model? If so, then the sole degree of freedom in the process model could be eliminated and optimization would not be needed!

Step 4. Not needed.

Step 5. Look at the total cost function, Equation (c). Observe that the cost function includes a constant term, K_3Q . If the total cost function is differentiated, the term K_3Q vanishes and thus K_3 does not enter into the determination of the optimal value for D . K_3 , however, contributes to the total cost.

Two approaches can be employed to solve for the optimal value of D : analytical or numerical. A simple problem has been formulated so that an analytical solution can

be obtained. Recall from calculus that if you differentiate the cost function with respect to D and equate the total derivative to zero

$$\frac{dC}{dD} = K_1 - \frac{K_2Q}{D^2} = 0 \quad (e)$$

you can obtain the optimal solution for D

$$D^{\text{opt}} = \sqrt{\frac{K_2Q}{K_1}} \quad (f)$$

Equation (f) was obtained without knowing specific numerical values for the parameters. If K_1 , K_2 , or Q change for one reason or another, then the calculation of the new value of D^{opt} is straightforward. Thus, the virtue of an analytical solution (versus a numerical one) is apparent.

Suppose you are given values of $K_1 = 1.0$, $K_2 = 10,000$, $K_3 = 4.0$, and $Q = 100,000$. Then D^{opt} from Equation (f) is 31,622.

You can also quickly verify for this problem that D^{opt} from Equation (f) minimizes the objective function by taking the second derivative of C and showing that it is positive. Equation (g) helps demonstrate the sufficient conditions for a minimum.

$$\frac{d^2C}{dD^2} = \frac{2K_2Q}{D^3} > 0 \quad (g)$$

Details concerning the necessary and sufficient conditions for minimization are presented in Chapter 4.

Another benefit of obtaining an analytical solution is that you can gain some insight into how production should be scheduled. For example, suppose the optimum number of production runs per year was 4.0 (25,000 units per run), and the projected demand for the product was doubled ($Q = 200,000$) for the next year. Using intuition you might decide to double the number of units produced (50,000 units) with 4.0 runs per year. However, as can be seen from the analytical solution, the new value of D^{opt} should be selected according to the square root of Q rather than the first power of Q . This relationship is known as the *economic order quantity* in inventory control and demonstrates some of the pitfalls that may result from making decisions by simple analogies or intuition.

We mentioned earlier that this problem was purposely designed so that an analytical solution could be obtained. Suppose now that the cost per run follows a non-linear function such as shown earlier in Figure E1.7. Let the cost vary as given by Equation (b), thus allowing for some economy of scale. Then the total cost function becomes

$$C = K_1D + \frac{K_2Q}{D} + \frac{K_4Q}{D^{1/2}} \quad (h)$$

After differentiation and equating the derivative to zero, you get

$$\frac{dC}{dD} = K_1 - \frac{K_2Q}{D^2} - \frac{K_4Q}{2D^{3/2}} = 0 \quad (i)$$

Note that Equation (i) is a rather complicated polynomial that cannot explicitly be solved for D^{opt} ; you have to resort to a numerical solution as discussed in Chapter 5.

A dichotomy arises in attempting to minimize function (*h*). You can either (1) minimize the cost function (*h*) directly or (2) find the roots of Equation (*i*). Which is the best procedure? In general it is easier to minimize *C* directly by a numerical method rather than take the derivative of *C*, equate it to zero, and solve the resulting nonlinear equation. This guideline also applies to functions of several variables.

The second derivative of Equation (*h*) is

$$\frac{d^2C}{dD^2} = \frac{2K_2Q}{D^3} + \frac{3K_4Q}{4D^{5/2}} \quad (j)$$

A numerical procedure to obtain D^{opt} directly from Equation (*d*) could also have been carried out by simply choosing values of *D* and computing the corresponding values of *C* from Equation (*d*) ($K_1 = 1.0$; $K_2 = 10,000$; $K_3 = 4.0$; $Q = 100,000$).

$D \times 10^{-3}$	10	20	30	40	50	60	70	80	90	100
$C \times 10^{-3}$	510	470	463	465	470	477	484	492	501	510

From the listed numerical data you can see that the function has a single minimum in the vicinity of $D = 20,000$ to $40,000$. Subsequent calculations in this range (on a finer scale) for *D* will yield a more precise value for D^{opt} .

Observe that the objective function value for $20 \leq D \leq 60$ does not vary significantly. However, not all functions behave like *C* in Equation (*d*)—some exhibit sharp changes in the objective function near the optimum.

Step 6. You should always be aware of the sensitivity of the optimal answer, that is, how much the optimal value of *C* changes when a variable such as *D* changes or a coefficient in the objective function changes. Parameter values usually contain errors or uncertainties. Information concerning the sensitivity of the optimum to changes or variations in a parameter is therefore very important in optimal process design. For some problems, a sensitivity analysis can be carried out analytically, but in others the sensitivity coefficients must be determined numerically.

In this example problem, we can analytically calculate the changes in C^{opt} in Equation (*d*) with respect to changes in the various cost parameters. Substitute D^{opt} from Equation (*f*) into the total cost function

$$C^{\text{opt}} = 2\sqrt{K_1K_2Q} + K_3Q \quad (k)$$

Next, take the partial derivatives of C^{opt} with respect to K_1 , K_2 , K_3 , and Q

$$\frac{\partial C^{\text{opt}}}{\partial K_1} = \sqrt{\frac{K_2Q}{K_1}} \quad (l1)$$

$$\frac{\partial C^{\text{opt}}}{\partial K_2} = \sqrt{\frac{K_1Q}{K_2}} \quad (l2)$$

$$\frac{\partial C^{\text{opt}}}{\partial K_3} = Q \quad (l3)$$

$$\frac{\partial C^{\text{opt}}}{\partial Q} = \sqrt{\frac{K_1K_2}{Q}} + K_3 \quad (l4)$$

Equations (l1) through (l4) are absolute sensitivity coefficients.

Similarly, we can develop expressions for the sensitivity of D^{opt} :

$$D^{\text{opt}} = \sqrt{\frac{K_2 Q}{K_1}} \quad (f)$$

$$\frac{\partial D^{\text{opt}}}{\partial K_1} = \frac{-1}{2K_1} \sqrt{\frac{K_2 Q}{K_1}} \quad (m1)$$

$$\frac{\partial D^{\text{opt}}}{\partial K_2} = \frac{1}{2K_2} \sqrt{\frac{K_2 Q}{K_1}} \quad (m2)$$

$$\frac{\partial D^{\text{opt}}}{\partial K_3} = 0 \quad (m3)$$

$$\frac{\partial D^{\text{opt}}}{\partial Q} = \frac{1}{2Q} \sqrt{\frac{K_2 Q}{K_1}} \quad (m4)$$

Suppose we now substitute numerical values for the constants in order to clarify how these sensitivity functions might be used. For

$$Q = 100,000 \quad K_1 = 1.0 \quad K_2 = 10,000 \quad K_3 = 4.0$$

then

$$D^{\text{opt}} = 31,622$$

$$C^{\text{opt}} = D^{\text{opt}} + \frac{10^9}{D^{\text{opt}}} + 400,000 = \$463,240$$

$$\frac{\partial C^{\text{opt}}}{\partial K_1} = 31,620 \quad \frac{\partial D^{\text{opt}}}{\partial K_1} = -15,810$$

$$\frac{\partial C^{\text{opt}}}{\partial K_2} = 3.162 \quad \frac{\partial D^{\text{opt}}}{\partial K_2} = 1.581$$

$$\frac{\partial C^{\text{opt}}}{\partial K_3} = 100,000 \quad \frac{\partial D^{\text{opt}}}{\partial K_3} = 0$$

$$\frac{\partial C^{\text{opt}}}{\partial Q} = 4.316 \quad \frac{\partial D^{\text{opt}}}{\partial Q} = 0.158$$

What can we conclude from the preceding numerical values? It appears that D^{opt} is extremely sensitive to K_1 , but not to Q . However, you must realize that a one-unit change in Q (100,000) is quite different from a one-unit change in K_1 (0.5). Therefore, in order to put the sensitivities on a more meaningful basis, you should compute the relative sensitivities: for example, the relative sensitivity of C^{opt} to K_1 is

$$S_{K_1}^C = \frac{\partial C^{\text{opt}}/C^{\text{opt}}}{\partial K_1/K_1} = \frac{\partial \ln C^{\text{opt}}}{\partial \ln K_1} = \sqrt{\frac{K_2 Q}{K_1}} \cdot \frac{K_1}{C^{\text{opt}}} = \frac{31,622(1.0)}{463,240} = 0.0683 \quad (n)$$

Application of the preceding idea for the other variables yields the other relative sensitivities for C^{opt} . Numerical values are

$$S_{K_3}^C = 0.863$$

$$S_{K_2}^C = 0.0683 \quad S_Q^C = 0.932$$

Changes in the parameters Q and K_3 have the largest relative influence on C^{opt} , significantly more than K_1 or K_2 . The relative sensitivities for D^{opt} are

$$S_{K_1}^D = -0.5 \quad S_{K_2}^D = S_Q^D = 0.5 \quad S_{K_3}^D = 0$$

so that all the parameters except for K_3 have the same influence (in terms of absolute value of fractional changes) on the optimum value of D .

For a problem for which we cannot obtain an analytical solution, you need to determine sensitivities numerically. You compute (1) the cost for the base case, that is, for a specified value of a parameter; (2) change each parameter separately (one at a time) by some arbitrarily small value, such as plus 1 percent or 10 percent, and then calculate the new cost. You might repeat the procedure for minus 1 percent or 10 percent. The variation of the parameter, of course, can be made arbitrarily small to approximate a differential; however, when the change approaches an infinitesimal value, the numerical error engendered may confound the calculations.

1.7 OBSTACLES TO OPTIMIZATION

If the objective function and constraints in an optimization problem are “nicely behaved,” optimization presents no great difficulty. In particular, if the objective function and constraints are all linear, a powerful method known as linear programming can be used to solve the optimization problem (refer to Chapter 7). For this specific type of problem it is known that a unique solution exists if any solution exists. However, most optimization problems in their natural formulation are not linear.

To make it possible to work with the relative simplicity of a linear problem, we often modify the mathematical description of the physical process so that it fits the available method of solution. Many persons employing computer codes for optimization do not fully appreciate the relation between the original problem and the problem being solved; the computer shows its neatly printed output with an authority that the reader feels unwilling, or unable, to question.

In this text we will discuss optimization problems based on behavior of physical systems that have a complicated objective function or constraints: for these problems some optimization procedures may be inappropriate and sometimes misleading. Often optimization problems exhibit one or more of the following characteristics, causing a failure in the calculation of the desired optimal solution:

1. The objective function or the constraint functions may have finite discontinuities in the continuous parameter values. For example, the price of a compressor or

heat exchanger may not change continuously as a function of variables such as size, pressure, temperature, and so on. Consequently, increasing the level of a parameter in some ranges has no effect on cost, whereas in other ranges a jump in cost occurs.

2. The objective function or the constraint functions may be nonlinear functions of the variables. When considering real process equipment, the existence of truly linear behavior and system behavior is somewhat of a rarity. This does not preclude the use of linear approximations, but the results of such approximations must be interpreted with considerable care.
3. The objective function or the constraint functions may be defined in terms of complicated interactions of the variables. A familiar case of interaction is the temperature and pressure dependence in the design of pressure vessels. For example, if the objective function is given as $f = 15.5x_1x_2^{1/2}$, the interaction between x_1 and x_2 precludes the determination of unique values of x_1 and x_2 . Many other more complicated and subtle interactions are common in engineering systems. The interaction prevents calculation of unique values of the variables at the optimum.
4. The objective function or the constraint functions may exhibit nearly "flat" behavior for some ranges of variables or exponential behavior for other ranges. This means that the value of the objective function or a constraint is not sensitive or is very sensitive, respectively, to changes in the value of the variables.
5. The objective function may exhibit many local optima, whereas the global optimum is sought. A solution to the optimization problem may be obtained that is less satisfactory than another solution elsewhere in the region. The better solution may be reached only by initiating the search for the optimum from a different starting point.

In subsequent chapters we will examine these obstacles and discuss some ways of mitigating such difficulties in performing optimization, but you should be aware these difficulties cannot always be alleviated.

REFERENCES

- Baumol, W. J. *Economic Theory and Operations Analysis*, 3rd ed. Prentice Hall, Englewood Cliffs, New Jersey (1972).
- Beveridge, G. S.; and R. S. Schechter. *Optimization: Theory and Practice*, McGraw-Hill, New York (1970).
- Romagnoli, J. A.; and M. C. Sanchez, *Data Processing and Reconciliation for Chemical Process Operation*. Academic Press, New York (1999).

SUPPLEMENTARY REFERENCES

- Hillier, F.; and G. Lieberman. *Introduction to Operations Research*. McGraw-Hill, New York (1995).
- Peters, M. S.; and K. D. Timmerhaus. *Plant Design and Economics for Chemical Engineers*. McGraw-Hill, New York (1991).

Reklaitis, G. V.; A. Ravindran, and K. M. Ragsdell. *Engineering Optimization*. John Wiley, New York (1983).

Rudd, D. F.; and C. C. Watson, *Strategy of Process Engineering*. John Wiley, New York (1968).

PROBLEMS

For each of the following six problems, formulate the objective function, the equality constraints (if any), and the inequality constraints (if any). Specify and list the independent variables, the number of degrees of freedom, and the coefficients in the optimization problem. Solve the problem using calculus as needed, and state the complete optimal solution values.

- 1.1 A poster is to contain 300 cm^2 of printed matter with margins of 6 cm at the top and bottom and 4 cm at each side. Find the overall dimensions that minimize the total area of the poster.
- 1.2 A box with a square base and open top is to hold 1000 cm^3 . Find the dimensions that require the least material (assume uniform thickness of material) to construct the box.
- 1.3 Find the area of the largest rectangle with its lower base on the x axis and whose corners are bounded at the top by the curve $y = 10 - x^2$.
- 1.4 Three points x are selected a distance h apart ($x_0, x_0 + h, x_0 + 2h$), with corresponding values f_0, f_1 , and f_2 . Find the maximum or minimum attained by a quadratic function passing through all three points. *Hint*: Find the coefficients of the quadratic function first.
- 1.5 Find the point on the curve $f = 2x^2 + 3x + 1$ nearest the origin.
- 1.6 Find the volume of the largest right circular cylinder that can be inscribed inside a sphere of radius R .
- 1.7 In a particular process the value of the product $f(x)$ is a function of the concentration x of ammonia expressed as a mole fraction. The following figure shows several values

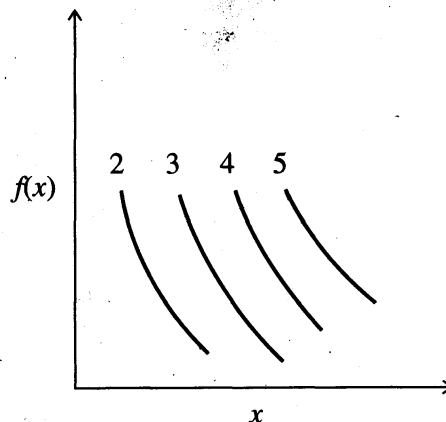


FIGURE P1.7

of $f(x)$. No units or values are designated for either of the axes. Duplicate the figure, and insert on the duplicate the constraint(s) involved in the problem by drawing very heavy lines or curves on the diagram.

- 1.8** A trucking company has borrowed \$600,000 for new equipment and is contemplating three kinds of trucks. Truck A costs \$10,000, truck B \$20,000, and truck C \$23,000. How many trucks of each kind should be ordered to obtain the greatest capacity in ton-miles per day based on the following data?

Truck A requires one driver per day and produces 2100 ton-miles per day.
 Truck B requires two drivers per day and produces 3600 ton-miles per day.
 Truck C requires two drivers per day and produces 3780 ton-miles per day.
 There is a limit of 30 trucks and 145 drivers.

Formulate a *complete* mathematical statement of the problem, and label each individual part, identifying the objective function and constraints with the correct units (\$, days, etc.). Make a list of the variables by names and symbol plus units. Do *not* solve.

- 1.9** In a rough preliminary design for a waste treatment plant the cost of the components are as follows (in order of operation)

1. Primary clarifier: $\$19.4 x_1^{-1.47}$
2. Trickling filter: $\$16.8 x_2^{-1.66}$
3. Activated sludge unit: $\$91.5 x_3^{-0.30}$

where the x 's are the fraction of the 5-day biochemical oxygen demand (BOD) exiting each respective unit in the process, that is, the exit concentrations of material to be removed.

The required removal in each unit should be adjusted so that the final exit concentration x_3 must be less than 0.05. Formulate (only) the optimization problem listing the objective function and constraints.

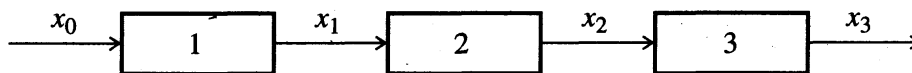


FIGURE P1.9

- 1.10** Examine the following optimization problem. State the total number of variables, and list them. State the number of independent variables, and list a set.

$$\text{Minimize: } f(x) = 4x_1 - x_2^2 - 12$$

$$\text{Subject to: } 25 - x_1^2 - x_2^2 = 0$$

$$10x_1 - x_1^2 + 10x_2 - x_2^2 - 34 \geq 0$$

$$(x_1 - 3)^2 + (x_2 - 1)^2 \geq 0$$

$$x_1, x_2 \geq 0$$

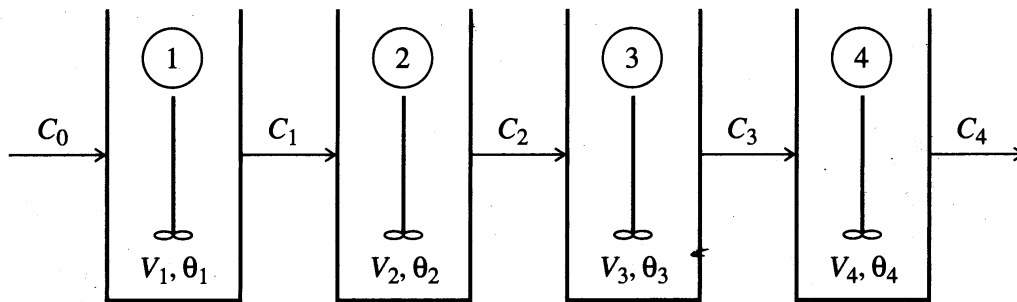


FIGURE P1.11

- 1.11** A series of four well-mixed reactors operate isothermally in the steady state. Examine the figure. All the tanks do not have the same volume, but the sum of $V_i = 20 \text{ m}^3$. The component whose concentration is designated by C reacts according to the following mechanism: $r = -kC^n$ in each tank.

Determine the values of the tank volumes (really residence times of the component) in each of the four tanks for steady-state operation with a fixed fluid flow rate of q so as to maximize the yield of product C_4 . Note $(V_i/q_i) = \theta_i$, the residence time. Use the following data for the coefficients in the problem

$$n = 2.5 \qquad k = 0.00625 \text{ [m}^3\text{/(kg mol)]}^{-1.5}\text{(s)}^{-1}$$

$$C_0 = 20 \text{ kg mol/m}^3 \qquad q = 71 \text{ m}^3\text{/h}$$

The units for k are fixed by the constant 0.00625.

List:

1. The objective function
2. The variables
3. The equality constraints
4. The inequality constraints

What are the independent variables? The dependent variables? Do not solve the problem, just set it up so it can be solved.

- 1.12** A certain gas contains moisture, which you need to remove by compression and cooling so that the gas will finally contain not more than 1% moisture (by volume). If the cost of the compression equipment is

$$\text{Cost in \$} = (\text{pressure in psi})^{1.40}$$

and the cost of the cooling equipment is

$$\text{Cost in \$} = (350 - \text{temperature in kelvin})^{1.9}$$

what is the best temperature to use?

Define the objective function, the independent and the dependent variables, and the constraints first. Then set this problem up, and list all of the steps to solve it. You

do not have to solve the final (nonlinear) equations you derive for T . *Hint*: The vapor pressure of water (p^*) is related to the temperature T in $^{\circ}\text{C}$ by Antoine's equation:

$$\log_{10} p^* = 8.10765 - \frac{1750.286}{235.0 + T}$$

- 1.13** The following problem is formulated as an optimization problem. A batch reactor operating over a 1-h period produces two products according to the parallel reaction mechanism: $A \rightarrow B$, $A \rightarrow C$. Both reactions are irreversible and first order in A and have rate constants given by

$$k_i = k_{i0} \exp \{E_i/RT\} \quad i = 1, 2$$

where $k_{10} = 10^6/\text{s}$

$$k_{20} = 5 \cdot 10^{11}/\text{s}$$

$$E_1 = 10,000 \text{ cal/gmol}$$

$$E_2 = 20,000 \text{ cal/gmol}$$

The objective is to find the temperature–time profile that maximizes the yield of B for operating temperatures below 282°F . The optimal control problem is therefore

$$\text{Maximize: } B(1.0)$$

$$\text{Subject to: } \frac{dA}{dt} = -(k_1 + k_2)A$$

$$\frac{dB}{dt} = k_1 A$$

$$A(0) = A_0$$

$$B(0) = 0$$

$$T \leq 282^{\circ}\text{F}$$

- What are the independent variables in the problem?
- What are the dependent variables in the problem?
- What are the equality constraints?
- What are the inequality constraints?
- What procedure would you recommend to solve the problem?

- 1.14** The computation of chemical equilibria can be posed as an optimization problem with linear side conditions. For any infinitesimal process in which the amounts of species present may be changed by either the transfer of species to or from a phase or by chemical reaction, the change in the Gibbs free energy is

$$dG = S dT + V dp + \sum_i \mu_i dn_i \quad (1)$$

Here G , S , T , and p are the Gibbs free energy, the entropy, the temperature, and the (total) pressure, respectively. The partial molal free energy of species number i is μ_i , and n_i is the number of moles of species number i in the system. If it is assumed that

the temperature and pressure are held constant during the process, dT and dp both vanish. If we now make changes in the n_i such that $dn_i = dk n_i$, so that the changes in the n_i are in the same proportion k , then, since G is an extensive quantity, we must have $dG = dkG$. This implies that

$$G = \sum_i \mu_i n_i \quad (2)$$

Comparison of Equations (1) and (2) shows that the chemical potentials are intensive quantities, that is, they do not depend on the amount of each species, because if all the n_i are increased in the same proportion at constant T and p , the μ_i must remain unchanged for G to increase in the same rate as the n_i . This invariance property of the μ_i is of the utmost importance in restricting the possible forms that the μ_i may take.

Equation (2) expresses the Gibbs free energy in terms of the mole numbers n_i , which appear both explicitly and implicitly (in the μ_i) on the right-hand side. The Gibbs free energy is a minimum when the system is at equilibrium. The basic problem, then, becomes that of finding that set of n_i that makes G a minimum.

- (a) Formulate in symbols the optimization problem using the previous notation with n_i^* being the number of moles of the compounds at equilibrium and M the number of elements present in the system. The initial number of moles of each compound is presumed to be known.
- (b) Introduce into the preceding formulation the quantities needed to solve the following problem:

Calculate the fraction of steam that is decomposed in the water-gas shift reaction



at $T = 1530^\circ\text{F}$ and $p = 10$ atm starting with 1 mol of H_2O and 1 mol of CO . Assume the mixture is an ideal gas. Do not solve the problem.

Hints: You can find (from a thermodynamics book) that the chemical potential can be written as

$$\mu_i = \mu_i^\circ + RT \ln p + RT \ln x_i = \mu_i^\circ + RT \ln p_i \quad (3)$$

where x_i = mole fraction of a compound in the gas phase

$$p_i = p x_i$$

$$\mu_{i,T}^\circ = (\Delta G_T^\circ)_i$$

$-(\Delta G_T^\circ) = RT \ln K_x$, with K_x being the equilibrium constant for the reaction.

1.15 For a two-stage adiabatic compressor where the gas is cooled to the inlet gas temperature between stages, the theoretical work is given by

$$W = \frac{k p_1 V_1}{k - 1} \left[\left(\frac{p_2}{p_1} \right)^{(k-1)/k} - 2 + \left(\frac{p_3}{p_2} \right)^{(k-1)/k} \right]$$

where $k = C_p/C_v$

p_1 = inlet pressure

p_2 = intermediate stage pressure

$$p_3 = \text{outlet pressure}$$

$$V_1 = \text{inlet volume}$$

We wish to optimize the intermediate pressure p_2 so that the work is a minimum. Show that if $p_1 = 1$ atm and $p_3 = 4$ atm, $p_2^{\text{opt}} = 2$ atm.

- 1.16** You are the manufacturer of PCl_3 , which you sell in barrels at a rate of P barrels per day. The cost per barrel produced is

$$C = 50 + 0.1P + 9000/P \text{ in dollars/barrel}$$

For example, for $P = 100$ barrels/day, $C = \$150/\text{barrel}$. The selling price per barrel is \$300. Determine

- The production level giving the minimum cost per barrel.
 - The production level which maximizes the profit per day.
 - The production level at zero profit.
 - Why are the answers in (a) and (b) different?
- 1.17** It is desired to cool a gas [$C_p = 0.3$ Btu/(lb)(°F)] from 195 to 90°F, using cooling water at 80°F. Water costs \$0.20/1000 ft³, and the annual fixed charges for the exchanger are \$0.50/ft² of inside surface, with a diameter of 0.0875 ft. The heat transfer coefficient is $U = 8$ Btu/(h)(ft²)(°F) for a gas rate of 3000 lb/h. Plot the annual cost of cooling water and fixed charges for the exchanger as a function of the outlet water temperature. What is the minimum total cost? How would you formulate the problem to obtain a more meaningful result? *Hint:* Which variable is the manipulated variable?
- 1.18** The total cost (in dollars per year) for pipeline installation and operation for an incompressible fluid can be expressed as follows:

$$C = C_1 D^{1.5} \cdot L + C_2 m \Delta p / \rho$$

where C_1 = the installed cost of the pipe per foot of length computed on an annual basis ($C_1 D^{1.5}$ is expressed in dollars per year per foot length, C_2 is based on \$0.05/kWh, 365 days/year and 60 percent pump efficiency).

D = diameter (to be optimized)

L = pipeline length = 100 miles

m = mass flow rate = 200,000 lb/h

$\Delta p = 2 \rho v^2 L / (D g_c) \cdot f$ = pressure drop, psi

ρ = density = 60 lb/ft³

v = velocity = $(4m) / (\rho \pi D^2)$

f = friction factor = $(0.046 \mu^{0.2}) / (D^{0.2} v^{0.2} \rho^{0.2})$

μ = viscosity = 1 cP

- Find general expressions for D^{opt} , v^{opt} , and C^{opt} .
- For $C_1 = 0.3$ (D expressed in inches for installed cost), calculate D^{opt} and v^{opt} for the following pairs of values of μ and ρ (watch your units!)

$$\mu = 0.2 \text{ cP}, 1 \text{ cP}, 10 \text{ cP}$$

$$\rho = 50 \text{ lb/ft}^3, 60 \text{ lb/ft}^3, 80 \text{ lb/ft}^3$$

- 1.19** Calculate the relative sensitivities of D^{opt} and C^{opt} in Problem 1.18 to changes in ρ , μ , m , and C_2 (cost of electricity). Use the base case parameters as given in Problem 1.18, with $C_1 = 0.3$.

Pose each of the following problems as an optimization problem. Include all of the features mentioned in connection with the first four steps of Table 1.1, but do not solve the problem.

- 1.20** A chemical manufacturing firm has discontinued production of a certain unprofitable product line. This has created considerable excess production capacity on the three existing batch production facilities that operate separately. Management is considering devoting this excess capacity to one or more of three new products; call them products 1, 2, and 3. The available capacity on the existing units which might limit output is summarized in the following table:

Unit	Available time (h/week)
A	20
B	10
C	5

Each of the three new products requires the following processing time for completion:

Unit	Productivity (h/batch)		
	Product 1	Product 2	Product 3
A	0.8	0.2	0.3
B	0.4	0.3	—
C	0.2	—	0.1

The sales department indicates that the sales potential for products 1 and 2 exceeds the maximum production rate and that the sales potential for product 3 is 20 batches per week. The profit per batch would be \$20, \$6, and \$8, respectively, on products 1, 2, and 3.

How much of each product should be produced to maximize profits of the company? Formulate the objective function and constraints, but do not solve.

- 1.21** You are asked to design an efficient treatment system for runoff from rainfall in an ethylene plant. The accompanying figure gives the general scheme to be used.

The rainfall frequency data for each recurrence interval fits an empirical equation in the form of

$$R = a + b(t)^2$$

where R = cumulative inches of rain during time t

t = time, h

a and b = constants that have to be determined by fitting the observed rainfall data

Four assumptions should be made:

1. The basin is empty at the beginning of the maximum intensity rain.
2. As soon as water starts to accumulate in the basin, the treatment system is started and water is pumped out of the basin.

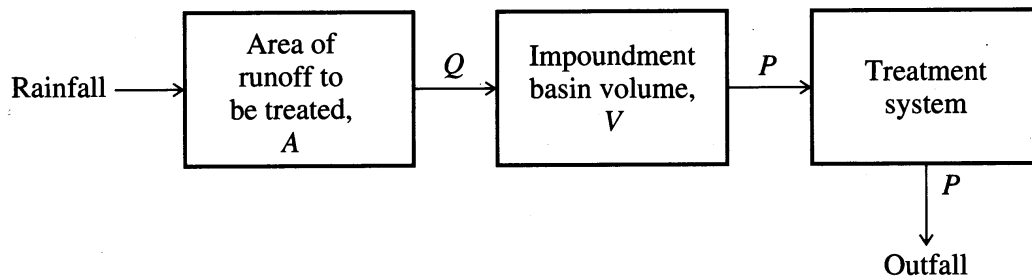


FIGURE P1.21

3. Stormwater is assumed to enter the basin as soon as it falls. (This is normally a good assumption since the rate at which water enters the basin is small relative to the rate at which it leaves the basin during a maximum intensity rain.)
4. All the rainfall becomes runoff.

The basin must not overflow so that any amount of water that would cause the basin to overflow must be pumped out and treated. What is the minimum pumping rate P required?

Other notation: Q = Volumetric flow rate of water entering basin
 P = Volumetric treatment rate in processing plant

- 1.22** Optimization of a distributed parameter system can be posed in various ways. An example is a packed, tubular reactor with radial diffusion. Assume a single reversible reaction takes place. To set up the problem as a nonlinear programming problem, write the appropriate balances (constraints) including initial and boundary conditions using the following notation:

x = Extent of reaction t = Time
 T = Dimensions temperature r = Dimensionless radial coordinate

Do the differential equations have to be expressed in the form of analytical solutions?

The objective function is to maximize the total conversion in the effluent from the reactor over the cross-sectional area at any instant of time. Keep in mind that the heat flux through the wall is subject to physical bounds.

- 1.23** Calculate a new expression for D^{opt} if $f = 0.005$ (rough pipe), independent of the Reynolds number. Compare your results with these from Problem 1.18 for $\mu = 1$ cP and $\rho = 60$ lb/ft³.
- 1.24** A shell-and-tube heat exchanger has a total cost of $C = \$7000 + \$250 D^{2.5}L + \$200 DL$, where D is the diameter (ft) and L is the length (ft). What is the absolute and the relative sensitivity of the total cost with respect to the diameter?

If an inequality constraint exists for the heat exchanger

$$20 \left(\frac{\pi D^2}{4} \right) L \geq 300$$

how must the sensitivity calculation be modified?

1.25 Empirical cost correlations for equipment are often of the following form:

$$\ln C = a_0 + a_1 \ln S + a_2 (\ln S)^2$$

where C is the base cost per unit and S is the size per unit. Obtain an analytical expression for the minimum cost in terms of S , and, if possible, find the expression that gives the value of S at the minimum cost. Also write down an analytical expression for the *relative* sensitivity of C with respect to S .

1.26 What are three major difficulties experienced in formulating optimization problems?

DEVELOPING MODELS FOR OPTIMIZATION

2.1 Classification of Models	41
2.2 How to Build a Model	46
2.3 Selecting Functions to Fit Empirical Data	48
2.4 Factorial Experimental Designs	62
2.5 Degrees of Freedom	66
2.6 Examples of Inequality and Equality Constraints in Models	69
References	73
Supplementary References	73
Problems	74

CONSTRAINTS IN OPTIMIZATION arise because a process must describe the physical bounds on the variables, empirical relations, and physical laws that apply to a specific problem, as mentioned in Section 1.4. How to develop models that take into account these constraints is the main focus of this chapter. Mathematical models are employed in all areas of science, engineering, and business to solve problems, design equipment, interpret data, and communicate information. Eykhoff (1974) defined a mathematical model as “a representation of the essential aspects of an existing system (or a system to be constructed) which presents knowledge of that system in a usable form.” For the purpose of optimization, we shall be concerned with developing quantitative expressions that will enable us to use mathematics and computer calculations to extract useful information. To optimize a process models may need to be developed for the objective function f , equality constraints \mathbf{g} , and inequality constraints \mathbf{h} .

Because a model is an abstraction, modeling allows us to avoid repetitive experimentation and measurements. Bear in mind, however, that a model only imitates reality and cannot incorporate all features of the real process being modeled. In the development of a model, you must decide what factors are relevant and how complex the model should be. For example, consider the following questions.

1. Should the process be modeled on a fundamental or empirical level, and what level of effort (time, expenses, manpower) is required for either approach?
2. Can the process be described adequately using physical principles?
3. What is the desired accuracy of the model, and how does its accuracy influence its ultimate use?
4. What measurements are available, and what data are available for model verification?
5. Is the process actually composed of smaller, simpler subsystems that can be more easily analyzed?

The answers to these questions depend on how the model is used. As the model of the process becomes more complex, optimization usually becomes more difficult.

In this chapter we will discuss several factors that need to be considered when constructing a process model. In addition, we will examine the use of optimization in estimating the values of unknown coefficients in models to yield a compact and reasonable representation of process data. Additional information can be found in textbooks specializing in mathematical modeling. To illustrate the need to develop models for optimization, consider the following example.

EXAMPLE 2.1 MODELING AND OPTIMIZING BLAST FURNACE OPERATION

Optimizing the operation of the blast furnace is important in every large-scale steel mill. A relatively large number of important variables (several of which cannot be measured) interact in this process in a highly complex manner, numerous constraints must be taken into account, and the age and efficiency of the plant significantly affect the optimum

operating point (Deitz, 1997). Consequently, a detailed examination of this problem demonstrates the considerations involved in mathematical modeling of a typical process.

The operation of a blast furnace is semicontinuous. The raw materials are iron ore containing roughly 20 to 60 percent iron as oxides and a variety of other metallic and nonmetallic oxides. These materials are combined with coke, which reacts to form blast furnace gas. Limestone is a flux that helps separate the impurities from the hot metal by influencing the pH. Apart from the blast furnace gas, which may serve as a heating medium in other processes, the output of the furnace consists of molten iron, which includes some impurities (notably carbon and phosphorus) that must be removed in the steelmaking process, and slag, which contains most of the impurities and is of little value. Operation of the blast furnace calls for determination of the amount of each ore, a production rate, and a mode of operation that will maximize the difference between the product value and the cost of producing the required quantity and quality of molten iron. Figure E2.1 shows the flow of materials in the blast furnace, which itself is part of a much larger mill. One ton of hot metal requires about 1.7 tons of iron-bearing materials, 0.5 to 0.65 tons of coke and other fuel, 0.25 tons of fluxes, and 1.8 to 2.0 tons of air. In addition, for each ton of hot metal produced, the process creates 0.2 to 0.4 tons of slag, 0.05 tons or less of flue dust, and 2.5 to 3.5 tons of blast furnace gases. The final product, hot metal, is about 93% iron, with other trace ingredients, including sulfur, silicon, phosphorus, and manganese. The process variables and conceptual models are identified in Figure E2.1 under the column "Process Analysis," which has categories for the objective function, equality constraints, and inequality constraints.

Objective function

To formulate the objective function, two categories of costs have to be considered:

1. Costs associated with the material flows (the input and output variables), such as the costs of purchased materials.
2. Costs associated with the operations related to the process variables in the model.

The terms that make up the objective function (to be maximized) are shown in Figure E.2.1. The profit of the blast furnace can be expressed as

$$f = \sum_{i=7}^8 c_i x_i - \sum_{i=1}^6 c_i x_i$$

Equality and inequality constraints

The next step in formulating the problem is to construct a mathematical model of the process by considering the fundamental chemical and physical phenomena and physical limitations that influence the process behavior. For the case of the blast furnace, typical features are

1. *Iron ore:* Ores of different grades are available in restricted quantities. Different ores have varying percentages of iron and different types and amounts of impurities. The proportion of each ore that occurs in the final hot metal is assumed to be fixed by its composition. For example, the amount of fine ore must be limited because too much can disrupt the flow of gas through the furnace and limit production.
2. *Coke:* The amount of coke that may be burned in any furnace is effectively limited by the furnace design, and the hot metal temperature is controlled by the amount

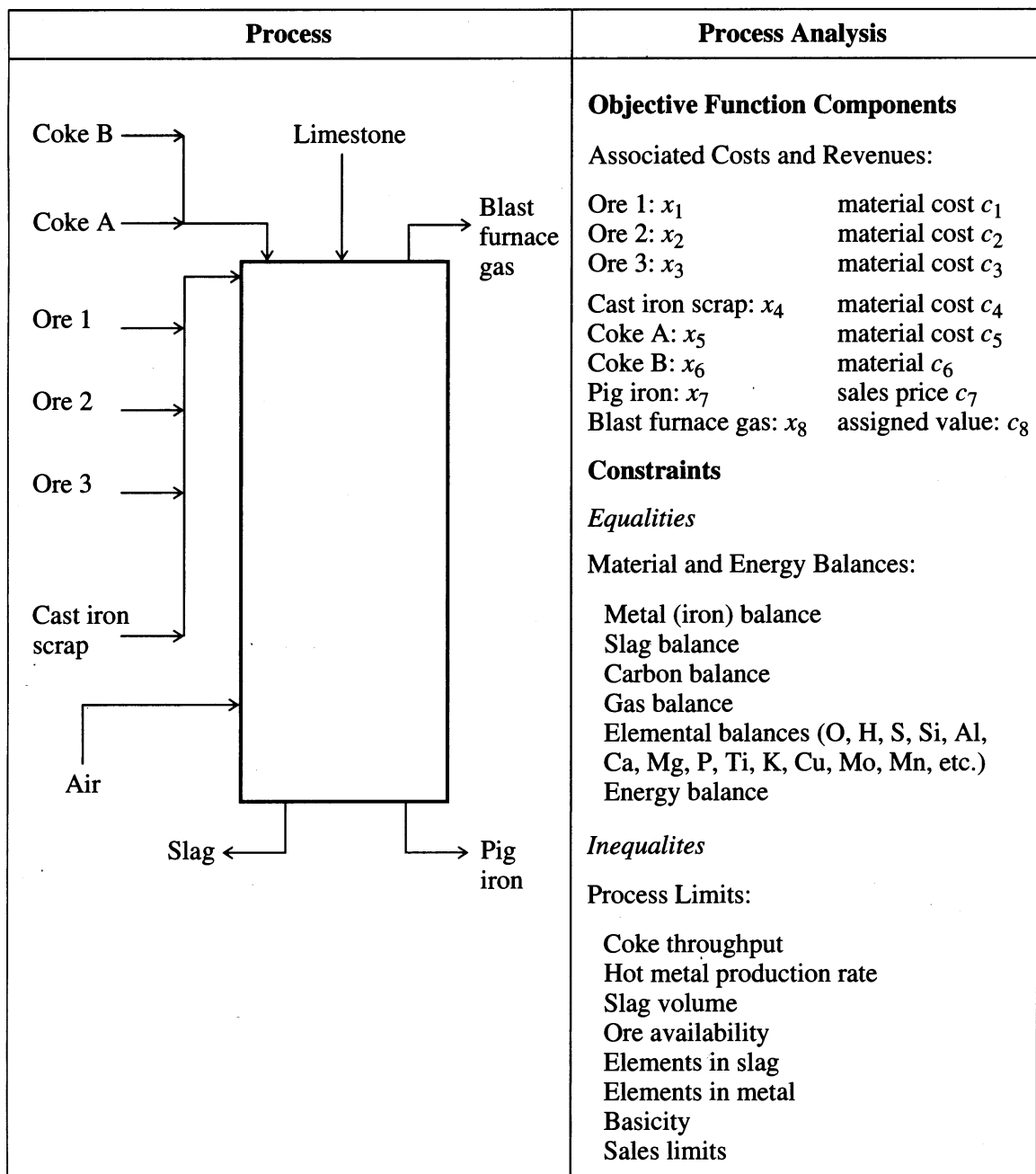


FIGURE E.2.1

Objective function components and types of constraints for a blast furnace.

of coke (or carbon). The coke consumption rate can be based on empirical relationships developed through regression of furnace data.

3. *Slag*: For technical reasons, the level of impurities in the slag must be controlled. There is an upper limit on the percentage of magnesium, upper and lower limits on the percentage of silicon and aluminum, and close limits on the “basicity” ratio $(CaO + MgO)/(SiO_2 + Al_2O_3)$. The basicity ratio controls the viscosity and melting point of the slag, which in turn affect the hearth temperature and grade of iron produced.

The basicity ratio can be expressed in terms of the blast furnace feeds x_i as follows:

$$\frac{\sum_{i=1}^4 w_{2i}x_i + \sum_{i=1}^4 w_{3i}x_i}{\sum_{i=1}^4 w_{4i}x_i + \sum_{i=1}^4 w_{5i}x_i}$$

where w_{2i} = weight fraction of CaO in feed i
 w_{3i} = weight fraction of MgO in feed i
 w_{4i} = weight fraction of SiO₂ in feed i
 w_{5i} = weight fraction of Al₂O₃ in feed i

4. *Phosphorus*: All phosphorus in the raw material finds its way into the molten metal. There is an upper limit on the phosphorus permitted, although precise quantities are sometimes prescribed. In general, it is cheaper to produce higher phosphorus iron, but more expensive to refine it.

From these and other considerations you can prepare:

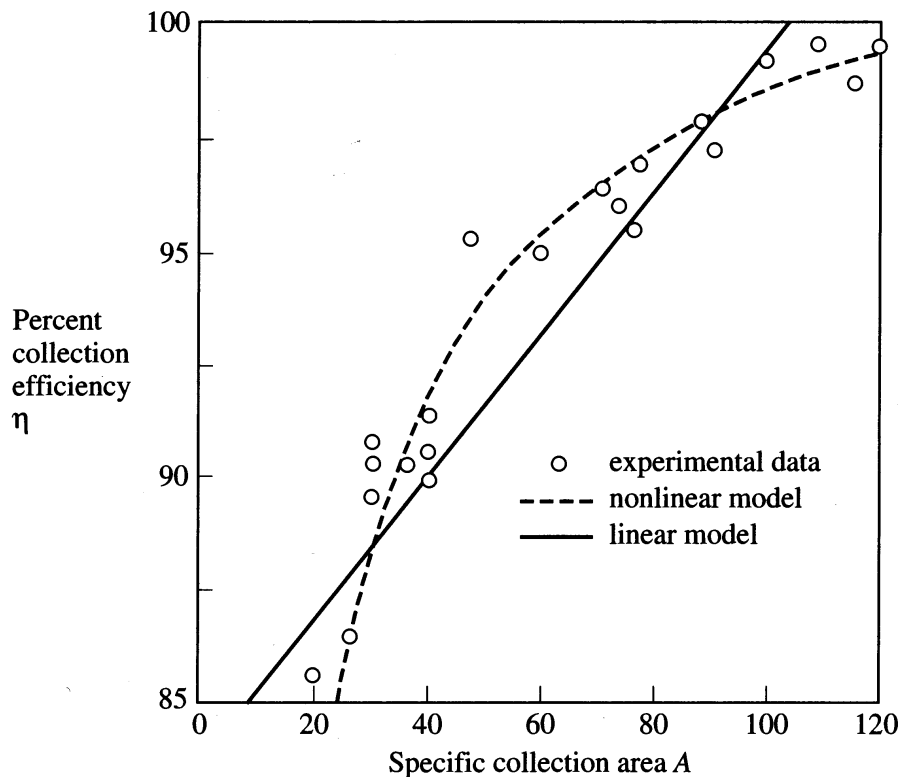
1. A set of input and output variables.
 2. A set of steady-state input–output material and energy balances (equality constraints).
 3. A set of explicit empirical relations (equality constraints).
 4. A set of restrictions (inequality constraints) on the input and output variables as indicated in Figure E.2.1.
-

2.1 CLASSIFICATION OF MODELS

Two general categories of models exist:

1. Those based on physical theory.
2. Those based on strictly empirical descriptions (so-called black box models).

Mathematical models based on physical and chemical laws (e.g., mass and energy balances, thermodynamics, chemical reaction kinetics) are frequently employed in optimization applications (refer to the examples in Chapters 11 through 16). These models are conceptually attractive because a general model for any system size can be developed even before the system is constructed. A detailed exposition of fundamental mathematical models in chemical engineering is beyond our scope here, although we present numerous examples of physiochemical models throughout the book, especially in Chapters 11 to 16. Empirical models, on the other hand, are attractive when a physical model cannot be developed due to limited time or resources. Input–output data are necessary in order to fit unknown coefficients in either type of the model.

**FIGURE E2.2**

ESP collection efficiency versus specific collection area for a linear model $\eta = 0.129A + 85.7$ and a nonlinear model $\eta = 100\{1 - [e^{-0.0264A}/(4.082 - 3.15 \times 10^{-6}A)]\}$.

EXAMPLE 2.2 MODELS OF AN ELECTROSTATIC PRECIPITATOR

A coal combustion pilot plant is used to obtain efficiency data on the collection of particulate matter by an electrostatic precipitator (ESP). The ESP performance is varied by changing the surface area of the collecting plates. Figure E2.2 shows the data collected to estimate the coefficients in a model to represent efficiency η as a function of the specific collection area A , measured as plate area/volumetric flow rate.

Two models of different complexity have been proposed to fit the performance data:

$$\text{Model 1: } \eta = b_1A + b_2$$

$$\text{Model 2: } \eta = 100 \left[1 - \frac{e^{-\gamma_1 A}}{\gamma_2 + \gamma_3 A} \right]$$

Model 1 is linear in the coefficients, and model 2 is nonlinear in the coefficients. The mathematical structure of model 2 has a fundamental basis that takes into account the physical characteristics of the particulate matter, including particle size and electrical properties, but we do not have the space to derive the equation here.

Which model is better?

Solution. The coefficients in the two models were fitted using MATLAB, yielding the following results:

$$\text{Model 1: } b_1 = 0.129 \quad b_2 = 85.7$$

$$\text{Model 2: } \gamma_1 = 0.0264 \quad \gamma_2 = 4.082 \quad \gamma_3 = -0.00000315$$

As can be seen in Figure E2.2, model 2 provides a better fit than model 1 over the range of areas A considered, but model 2 may present some difficulties when used as a constraint inserted into an optimization code.

The electrostatic precipitator in Example 2.2 is typical of industrial processes; the operation of most process equipment is so complicated that application of fundamental physical laws may not produce a suitable model. For example, thermodynamic or chemical kinetics data may be required in such a model but may not be available. On the other hand, although the development of black box models may require less effort and the resulting models may be simpler in form, empirical models are usually only relevant for restricted ranges of operation and scale-up. Thus, a model such as ESP model 1 might need to be completely reformulated for a different size range of particulate matter or for a different type of coal. You might have to use a series of black box models to achieve suitable accuracy for different operating conditions.

In addition to classifying models as theoretically based versus empirical, we can generally group models according to the following types:

Linear versus nonlinear.

Steady state versus unsteady state.

Lumped parameter versus distributed parameter.

Continuous versus discrete variables.

Linear versus nonlinear

Linear models exhibit the important property of superposition; nonlinear ones do not. Equations (and hence models) are linear if the dependent variables or their derivatives appear only to the first power: otherwise they are nonlinear. In practice the ability to use linear models is of great significance because they are an order of magnitude easier to manipulate and solve than nonlinear ones.

To test for the linearity of a model, examine the equation(s) that represents the process. If any one term is nonlinear, the model itself is nonlinear. By implication, the process is nonlinear.

Examine models 1 and 2 for the electrostatic precipitator. Is model 1 linear in A ? Model 2? The superposition test in each case is: Does

$$J(ax_1 + bx_2) = aJ(x_1) + bJ(x_2) \quad (2.1a)$$

and

$$J(kx) = kJ(x) \quad (2.1b)$$

where J = any operator contained in the model such as square, differentiation, and so on.

k = a constant

x_1 and x_2 = variables

ESP model 1 is linear in A

$$J(b_1A + b_2) = b_1J(A) + b_2$$

but ESP model 2 is nonlinear because

$$\left(\frac{e^{-\gamma_1(A_1 + A_2)}}{\gamma_2 + \gamma_3(A_1 + A_2)} \right) \neq \left(\frac{e^{-\gamma_1 A_1}}{\gamma_2 + \gamma_3 A_1} \right) + \left(\frac{e^{-\gamma_1 A_2}}{\gamma_2 + \gamma_3 A_2} \right)$$

Steady state versus unsteady state

Other synonyms for steady state are time-invariant, static, or stationary. These terms refer to a process in which the values of the dependent variables remain constant with respect to time. Unsteady state processes are also called nonsteady state, transient, or dynamic and represent the situation when the process-dependent variables change with time. A typical example of an unsteady state process is the operation of a batch distillation column, which would exhibit a time-varying product composition. A transient model reduces to a steady state model when $\partial/\partial t = 0$. Most optimization problems treated in this book are based on steady state models. Optimization problems involving dynamic models usually pertain to “optimal control” or real-time optimization problems (see Chapter 16)

Distributed versus lumped parameters

Briefly, a lumped parameter representation means that spatial variations are ignored and that the various properties and the state of the system can be considered homogeneous throughout the entire volume. A distributed parameter representation, on the other hand, takes into account detailed variations in behavior from point to point throughout the system. In Figure 2.1, compare these definitions for a well-stirred reactor and a tubular reactor with axial flow. In the first case, we assume that mixing is complete so no concentration or temperature gradient occurs in the reactor, hence a lumped parameter mathematical model would be appropriate. In contrast, the tubular reactor has concentration or temperature variations along the axial direction and perhaps in the radial direction, hence a distributed parameter model would be required. All real systems are, of course, distributed because some variations of states occur throughout them. Because the spatial variations often are relatively small, they may be ignored, leading to a lumped approximation. If both spatial and transient characteristics are to be included in a model, a partial differential equation or a series of stages is required to describe the process behavior.

It is not easy to determine whether lumping in a process model is a valid technique for representing the process. A good rule of thumb is that if the response is

essentially the same at all points in the process, then the model can be lumped as a single unit. If the response shows significant instantaneous differences in any direction along the vessel, then the problem should be treated using an appropriate differential equation or series of compartments. In an optimization problem it is desirable to simplify a distributed model by using an equivalent lumped parameter system, although you must be careful to avoid masking the salient features of the distributed element (hence building an inadequate model). In this text, we will mainly consider optimization techniques applied to lumped systems.

Continuous versus discrete variables

Continuous variables can assume any value within an interval; discrete variables can take only distinct values. An example of a discrete variable is one that assumes integer values only. Often in chemical engineering discrete variables and continuous variables occur simultaneously in a problem. If you wish to optimize a compressor system, for example, you must select the number of compressor stages (an integer) in addition to the suction and production pressure of each stage (positive continuous variables). Optimization problems without discrete variables are far easier to solve than those with even one discrete variable. Refer to Chapter 9 for more information about the effect of discrete variables in optimization.

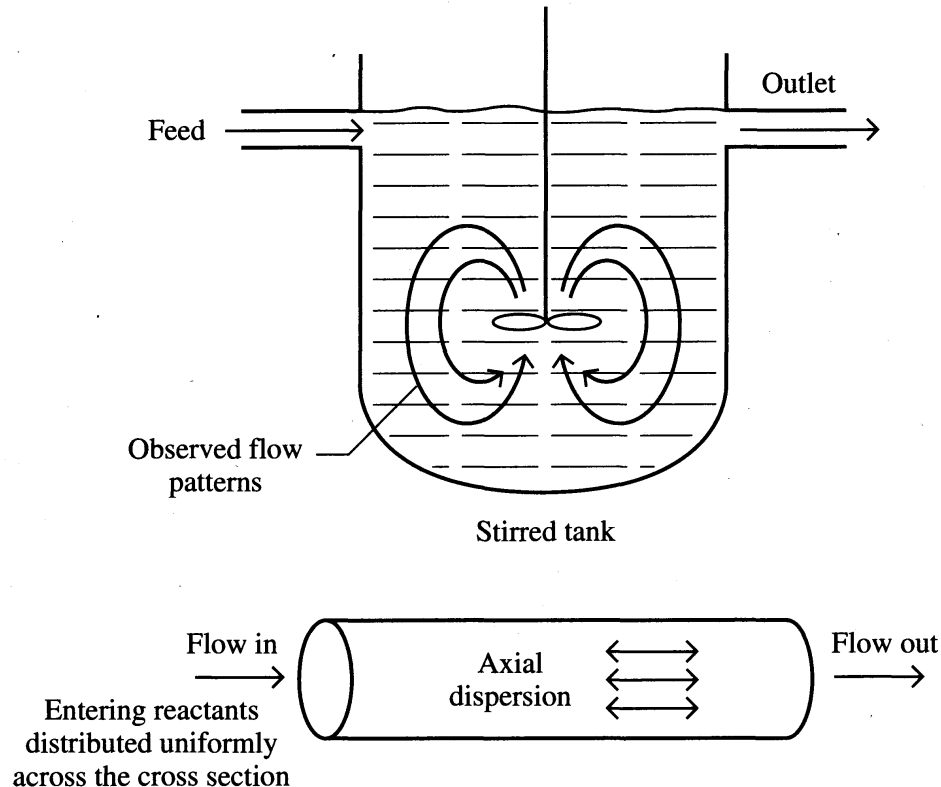


FIGURE 2.1

Flow patterns in a stirred tank (lumped parameter system) and a tubular reactor (distributed parameter system).

An engineer typically strives to treat discrete variables as continuous even at the cost of achieving a suboptimal solution when the continuous variable is rounded off. Consider the variation of the cost of insulation of various thickness as shown in Figure E1.1. Although insulation is only available in 0.5-in. increments, continuous approximation for the thickness can be used to facilitate the solution to this optimization problem.

2.2 HOW TO BUILD A MODEL

For convenience of presentation, model building can be divided into four phases: (1) problem definition and formulation, (2) preliminary and detailed analysis, (3) evaluation, and (4) interpretation application. Keep in mind that model building is an iterative procedure. Figure 2.2 summarizes the activities to be carried out,

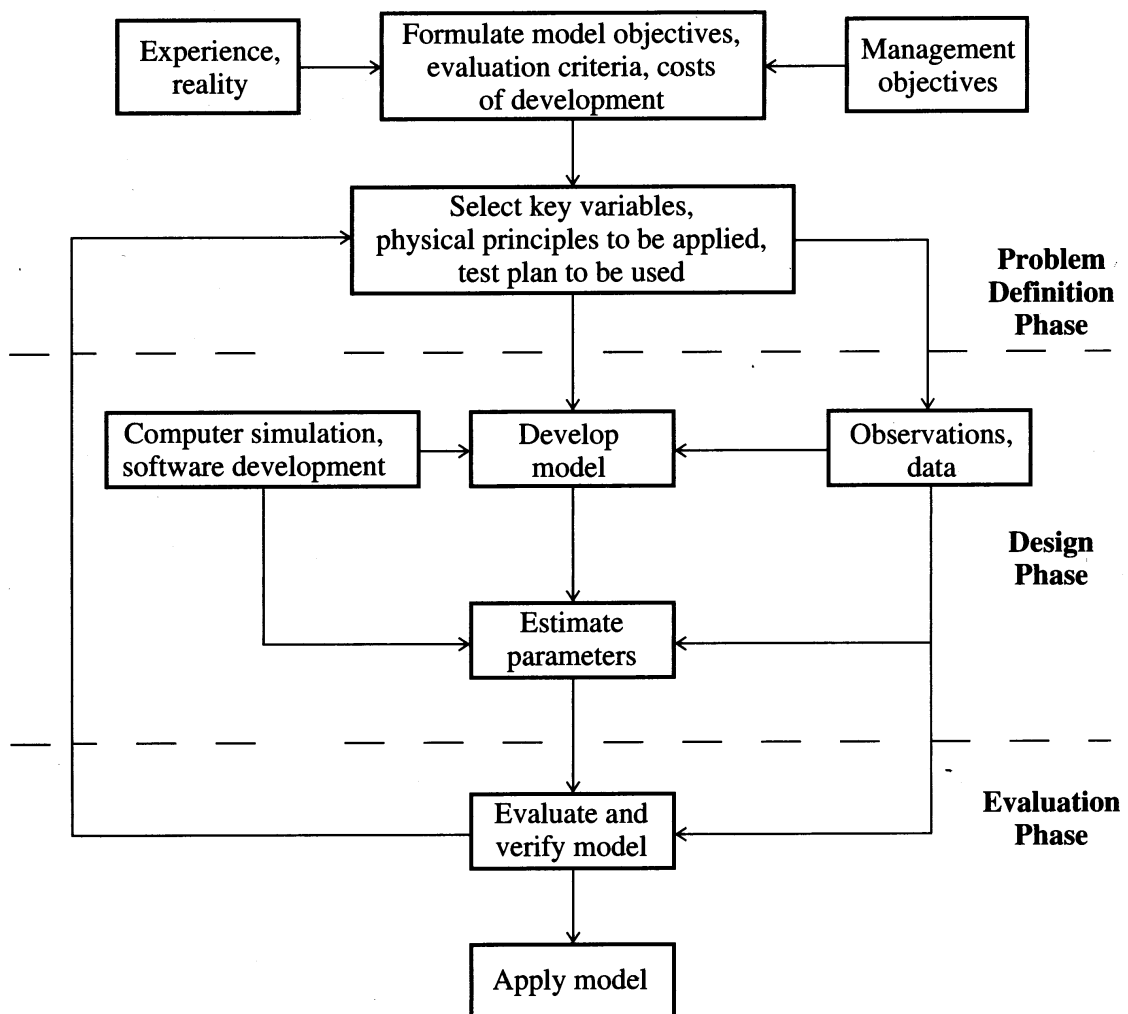


FIGURE 2.2
Major activities in model building prior to application.

which are discussed in detail later on. The content of this section is quite limited in scope; before actually embarking on a comprehensive model development program, consult textbooks on modeling (see References).

Problem definition and formulation phase

In this phase the problem is defined and the important elements that pertain to the problem and its solution are identified. The degree of accuracy needed in the model and the model's potential uses must be determined. To evaluate the structure and complexity of the model, ascertain

1. The number of independent variables to be included in the model.
2. The number of independent equations required to describe the system (sometimes called the "order" of the model).
3. The number of unknown parameters in the model.

In the previous section we addressed some of these issues in the context of physical versus empirical models. These issues are also intertwined with the question of model verification: what kinds of data are available for determining that the model is a valid description of the process? Model building is an iterative process, as shown by the recycling of information in Figure 2.2.

Before carrying out the actual modeling, it is important to evaluate the economic justification for (and benefits of) the modeling effort and the capability of support staff for carrying out such a project. Primarily, determine that a successfully developed model will indeed help solve the optimization problem.

Design phase

The design phase includes specification of the information content, general description of the programming logic and algorithms necessary to develop and employ a useful model, formulation of the mathematical description of such a model, and simulation of the model. First, define the input and output variables, and determine what the "system" and the "environment" are. Also, select the specific mathematical representation(s) to be used in the model, as well as the assumptions and limitations of the model resulting from its translation into computer code. Computer implementation of the model requires that you verify the availability and adequacy of computer hardware and software, specify computer input-output media, develop program logic and flowsheets, and define program modules and their structural relationships. Use of existing subroutines and databases saves you time but can complicate an optimization problem for the reasons explained in Chapter 15.

Evaluation phase

This phase is intended as a final check of the model as a whole. Testing of individual model elements should be conducted during earlier phases. Evaluation of the model is carried out according to the evaluation criteria and test plan established in the problem definition phase. Next, carry out sensitivity testing of the model inputs

and parameters, and determine if the apparent relationships are physically meaningful. Use actual data in the model when possible. This step is also referred to as diagnostic checking and may entail statistical analysis of the fitted parameters (Box et al., 1978).

Model validation requires confirming logic, assumptions, and behavior. These tasks involve comparison with historical input–output data, or data in the literature, comparison with pilot plant performance, and simulation. In general, data used in formulating a model should not be used to validate it if at all possible. Because model evaluation involves multiple criteria, it is helpful to find an expert opinion in the verification of models, that is, what do people think who know about the process being modeled?

No single validation procedure is appropriate for all models. Nevertheless, it is appropriate to ask the question: What do you want the model to do? In the best of all possible worlds, you want the model to predict the desired process performance with suitable accuracy, but this is often an elusive goal.

2.3 SELECTING FUNCTIONS TO FIT EMPIRICAL DATA

A model relates the output (the dependent variable or variables) to the independent variable(s). Each equation in the model usually includes one or more coefficients that are presumed constant. The term *parameter* as used here means coefficient and possibly input or initial condition. With the help of experimental data, we can determine the *form* of the model and subsequently (or simultaneously) estimate the value of some or all of the parameters in the model.

2.3.1 How to Determine the Form of a Model

Models can be written in a variety of mathematical forms. Figure 2.3 shows a few of the possibilities, some of which were already illustrated in Section 2.1. This section focuses on the simplest case, namely models composed of algebraic equations, which constitute the bulk of the equality constraints in process optimization. Emphasis here is on estimating the coefficients in simple models and not on the complexity of the model.

Selection of the form of an empirical model requires judgment as well as some skill in recognizing how response patterns match possible algebraic functions. Optimization methods can help in the selection of the model structure as well as in the estimation of the unknown coefficients. If you can specify a quantitative criterion that defines what “best” represents the data, then the model can be improved by adjusting its form to improve the value of the criterion. The best model presumably exhibits the least error between actual data and the predicted response in some sense.

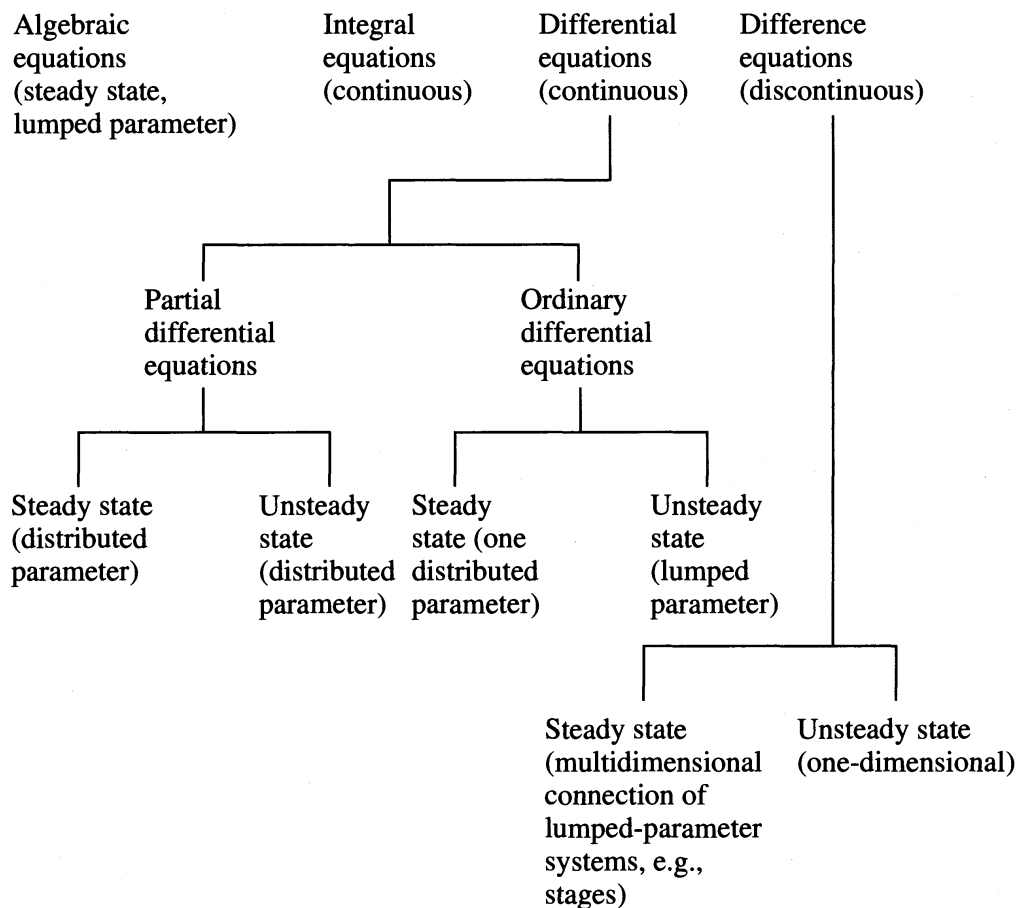


FIGURE 2.3
Typical mathematical forms of models.

Typical relations for empirical models might be

$$y = a_0 + a_1x_1 + a_2x_2 + \cdots \quad \text{linear in the variables and coefficients}$$

$$y = a_0 + a_{11}x_1^2 + a_{12}x_1x_2 + \cdots \quad \text{linear in the coefficients, nonlinear in the variables } (x_1, x_2)$$

$$G(s) = \frac{1}{a_0 + a_1s + a_2s^2} \quad \text{nonlinear in all the coefficients}$$

$$\bullet \text{Nu} = a(\text{Re})^b \quad \text{nonlinear in the coefficient } b$$

(Nu: Nusselt number; Re: Reynolds number)

When the model is linear in the coefficients, they can be estimated by a procedure called *linear regression*. If the model is nonlinear in the coefficients, estimating them is referred to as *nonlinear regression*. In either case, the simplest adequate model (with the fewest number of coefficients) should be used.

Graphical presentation of data assists in determining the form of the function of a single variable (or two variables). The response y versus the independent variable x can be plotted and the resulting form of the model evaluated visually. Figure 2.4 shows experimental heat transfer data plotted on log–log coordinates. The plot

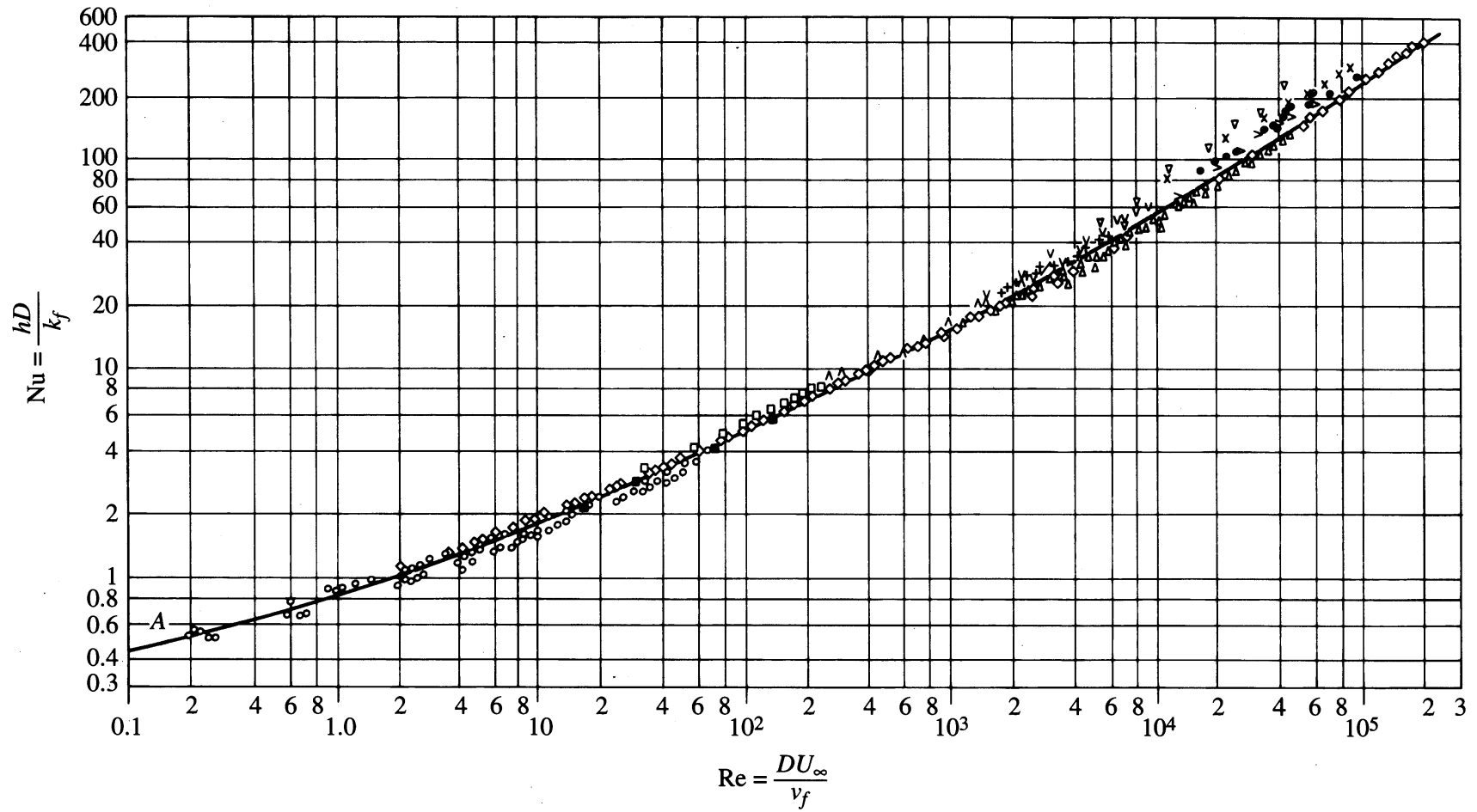


FIGURE 2.4

Average Nusselt number (Nu) versus Reynolds number (Re) for a circular cylinder in air, placed normal to the flow (McAdams, 1954, with permission from McGraw-Hill Companies).

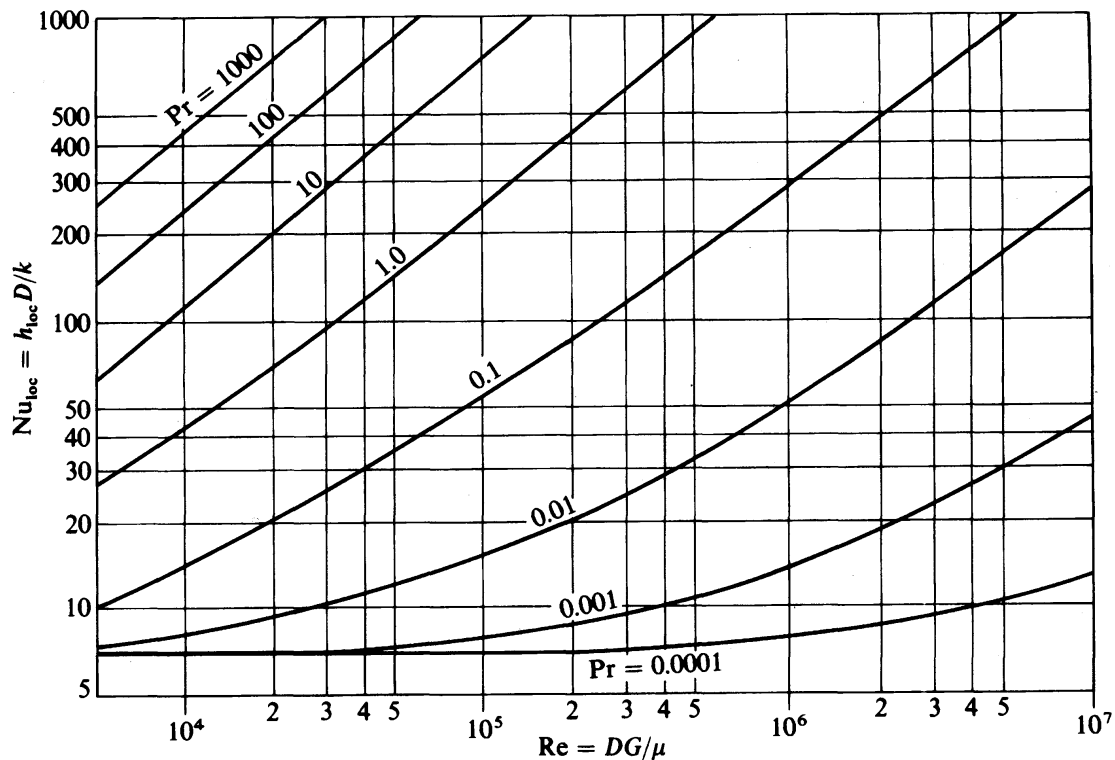


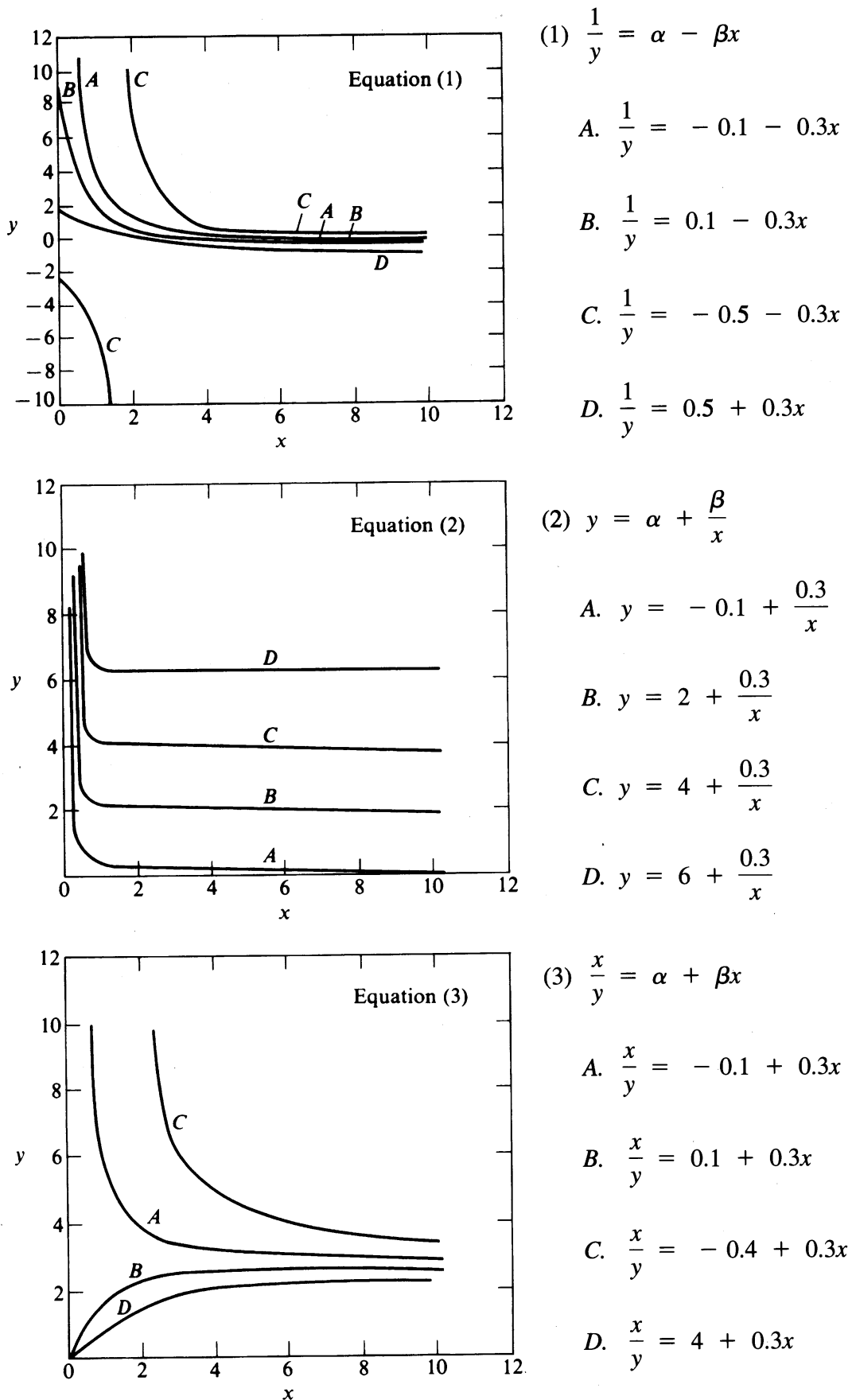
FIGURE 2.5

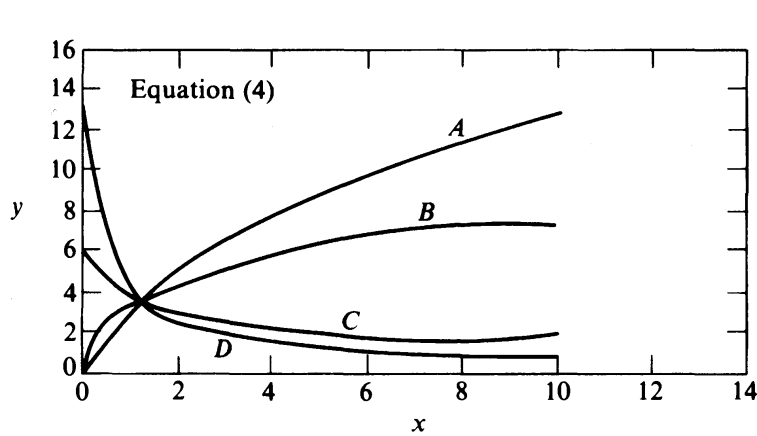
Predicted Nusselt numbers for turbulent flow with constant wall heat flux (*adapted with permission from John Wiley and Sons from Bird et al., 1964*). Abbreviations: Nu = Nusselt number; Re = Reynolds number; Pr = Prandtl number.

appears to be approximately linear over wide ranges of the Reynolds number (Re). A straight line in Figure 2.4 would correspond to $\log \text{Nu} = \log a + b \log \text{Re}$ or $\text{Nu} = a(\text{Re})^b$. Observe the scatter of experimental data in Figure 2.4, especially for large values of the Re.

If two independent variables are involved in the model, plots such as those shown in Figure 2.5 can be of assistance; in this case the second independent variable becomes a parameter that is held constant at various levels. Figure 2.6 shows a variety of nonlinear functions and their associated plots. These plots can assist in selecting relations for nonlinear functions of y versus x . Empirical functions of more than two variables must be built up (or pruned) step by step to avoid including an excessive number of irrelevant variables or missing an important one. Refer to Section 2.4 for suitable procedures.

Now let us review an example for selecting the form of a model to fit experimental data.

**FIGURE 2.6**Functions of a single variable x and their corresponding trajectories. (*Continues*)



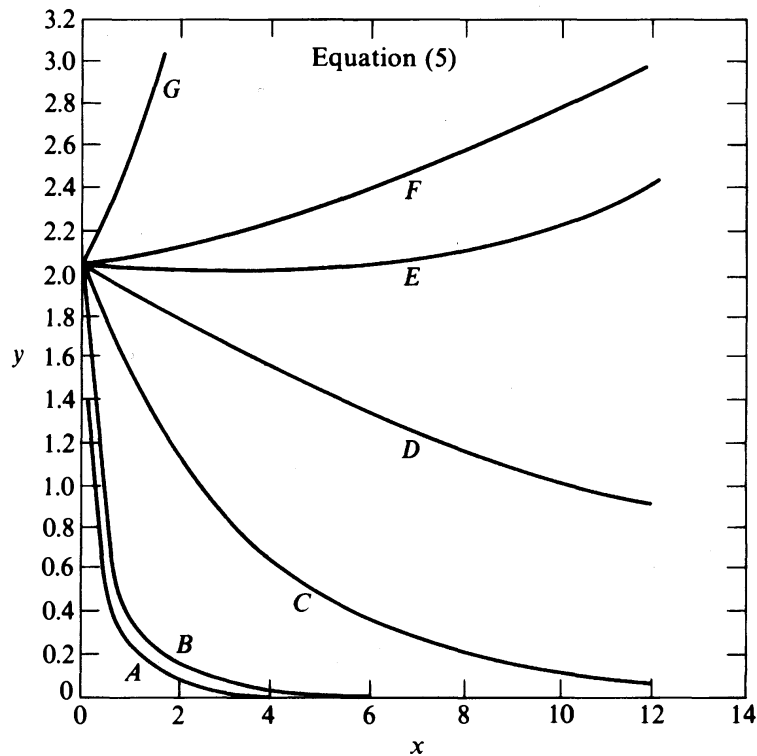
(4) $y = \alpha x^\beta$

A. $y = 4x^{0.5}$

B. $y = 4x^{0.3}$

C. $y = 4x^{-0.3}$

D. $y = 4x^{-0.5}$



(5) $y = \alpha\beta^x$

A. $y = 2(0.2)^x$

B. $y = 2(0.3)^x$

C. $y = 2(0.8)^x$

D. $y = 2(0.95)^x$

E. $y = 2(1.02)^x$

F. $y = 2(1.04)^x$

G. $y = 2(1.3)^x$

FIGURE 2.6 (continued)

EXAMPLE 2.3 ANALYSIS OF THE HEAT TRANSFER COEFFICIENT

Suppose the overall heat transfer coefficient of a shell-and-tube heat exchanger is calculated daily as a function of the flow rates in both the shell and tube sides (w_s and w_t , respectively). U has the units of $\text{Btu}/(\text{h})(^\circ\text{F})(\text{ft}^2)$, and w_s and w_t are in lb/h . Figures E2.3a and E2.3b illustrate the measured data. Determine the form of a semiempirical model of U versus w_s and w_t based on physical analysis.

Solution. You could elect to simply fit U as a polynomial function of w_s and w_t ; there appears to be very little effect of w_s on U , but U appears to vary linearly with w_t (except at the upper range of w_t where it begins to level off). A more quantitative approach

can be based on a physical analysis of the exchanger. First determine why w_s has no effect on U . This result can be explained by the formula for the overall heat transfer coefficient

$$\frac{1}{U} = \frac{1}{h_s} + \frac{1}{h_t} + \frac{1}{h_f} \quad (a)$$

where h_s = the shell heat transfer coefficient
 h_t = the tube side heat transfer coefficient
 h_f = the fouling coefficient

If h_t is small and h_s is large, U is dominated by h_t , hence changes in w_s have little effect, as shown in Figure E2.3a.

Next examine the data for U versus w_t in the context of Figure 2.6. For a reasonable range of w_t , the pattern is similar to curve D in Equation (3) where

$$\frac{x}{y} = \alpha + \beta x \quad (b)$$

which can also be written as

$$\frac{1}{y} = \frac{\alpha}{x} + \beta \quad (c)$$

Note the similarity between Equations (c) and (a), where $x = h_t$ and $y = U$. From a standard heat transfer coefficient correlation (Gebhart, 1971), you can find that h_t also varies according to $K_t w_t^{0.8}$, where K_t is a coefficient that depends on the fluid physical properties and the exchanger geometry. If we lump $1/h_s$ and $1/h_f$ together into one constant $1/h_{sf}$, the semiempirical model becomes

$$\frac{1}{U} = \frac{1}{h_{sf}} + \frac{1}{K_t w_t^{0.8}}$$

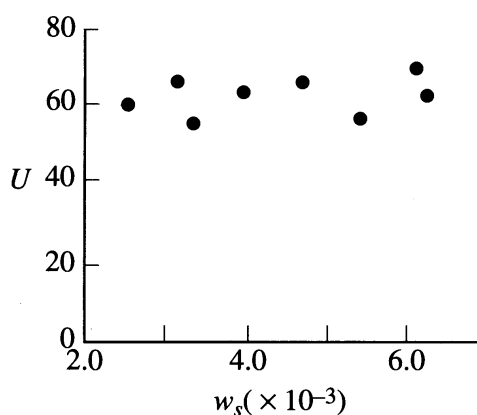


FIGURE E2.3a
 Variation of overall heat transfer coefficient with shell-side flow rate $w_s = 8000$.

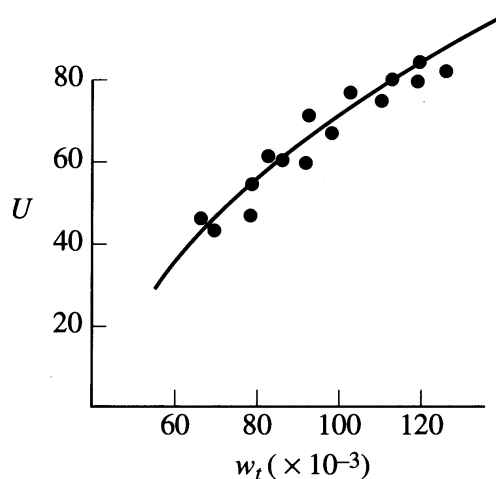


FIGURE E2.3b
 Variation of overall heat transfer coefficient with tube-side flow rate w_t for $w_s = 4000$.

or

$$U = \frac{h_{sf}K_i w_i^{0.8}}{K_i w_i^{0.8} + h_{sf}} \quad (d)$$

The line in Figure E2.3b shows how well Equation (d) fits the data.

In the previous examples and figures we indicated that functions for two independent variables can be selected. When three (or more) independent variables occur, advanced analysis tools, such as experimental design (see Section 2.4) or principal component analysis (Jackson, 1991), are required to determine the structure of the model.

Once the form of the model is selected, even when it involves more than two independent variables, fitting the unknown coefficients in the model using linear or nonlinear regression is reasonably straightforward. We discuss methods of fitting coefficients in the next section.

2.3.2 Fitting Models by Least Squares

This section describes the basic idea of least squares estimation, which is used to calculate the values of the coefficients in a model from experimental data. In estimating the values of coefficients for either an empirical or theoretically based model, keep in mind that the number of data sets must be equal to or greater than the number of coefficients in the model. For example, with three data points of y versus x , you can estimate at most the values of three coefficients. Examine Figure 2.7. A straight line might represent the three points adequately, but the data can be fitted exactly using a quadratic model

$$y = \beta_0 + \beta_1 x + \beta_2 x^2 \quad (2.2)$$

By introducing the values of a data point (Y_1, x_1) into Equation 2.2, you obtain one equation of Y_1 as a function of three unknown coefficients. The set of three data points therefore yields three linear equations in three unknowns (the coefficients) that can be solved easily.

To compensate for the errors involved in experimental data, the number of data sets should be greater than the number of coefficients p in the model. Least squares is just the application of optimization to obtain the “best” solution of the equations, meaning that the sum of the squares of the errors between the predicted and the experimental values of the dependent variable y for each data point x is minimized. Consider a general algebraic model that is linear in the coefficients.

$$y = \sum_{j=1}^p \beta_j x_j \quad (2.3)$$

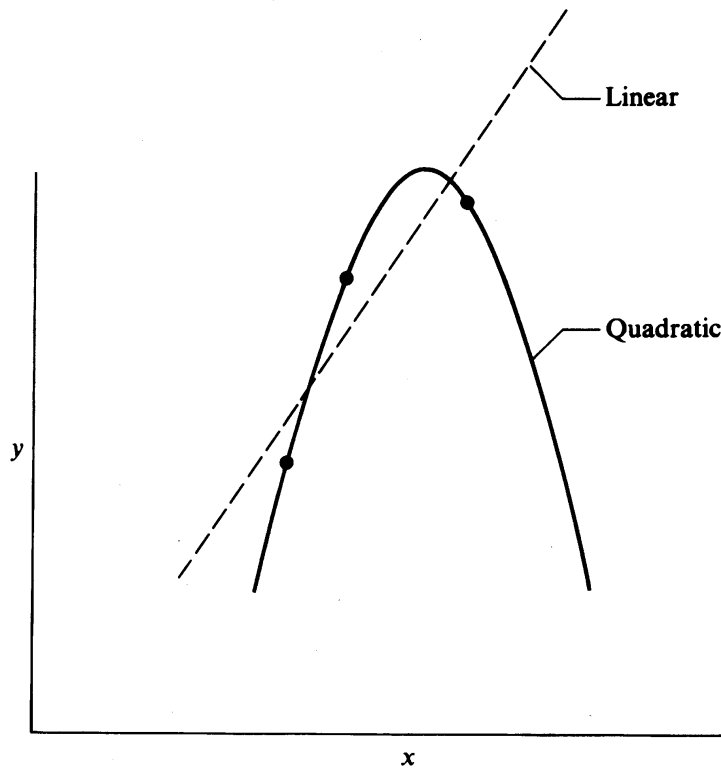


FIGURE 2.7
Linear versus quadratic fit for three data points.

There are p independent variables $x_j, j = 1, \dots, p$. Independent here means controllable or adjustable, not functionally independent. Equation (2.3) is linear with respect to the β_j , but x_j can be nonlinear. Keep in mind, however, that the values of x_j (based on the input data) are just numbers that are substituted prior to solving for the estimates $\hat{\beta}_j$, hence nonlinear functions of x_j in the model are of no concern. For example, if the model is a quadratic function,

$$y = \beta_1 + \beta_2 x + \beta_3 x^2$$

we specify

$$x_1 = 1$$

$$x_2 = x$$

$$x_3 = x^2$$

and the general structure of Equation (2.3) is satisfied. In reading Section 2.4 you will learn that special care must be taken in collecting values of x to avoid a high degree of correlation between the x_i 's.

Introduction of Equation (2.3) into a sum-of-squares error objective function gives

$$f = \sum_{i=1}^n \left(Y_i - \sum_{j=1}^p \beta_j x_{ij} \right)^2 \quad (2.4)$$

The objective function to be minimized is

$$f = \boldsymbol{\varepsilon}^T \boldsymbol{\varepsilon} = (\mathbf{Y} - \mathbf{x}\boldsymbol{\beta})^T (\mathbf{Y} - \mathbf{x}\boldsymbol{\beta}) \quad (2.7)$$

Equations 2.5 can then be expressed as

$$\mathbf{x}^T \mathbf{x} \hat{\boldsymbol{\beta}} = \mathbf{x}^T \mathbf{Y} \quad (2.8)$$

which has the formal solution via matrix algebra

$$\hat{\boldsymbol{\beta}} = (\mathbf{x}^T \mathbf{x})^{-1} \mathbf{x}^T \mathbf{Y} \quad (2.9)$$

Statistical packages and spreadsheets solve the simultaneous equations in (2.8) to estimate $\hat{\boldsymbol{\beta}}$ rather than computing the matrix inverse in Equation (2.9).

The next two examples illustrate the application of Equation 2.9 to fit coefficients in an objective function. The same procedure is used to fit coefficients in constraint models.

EXAMPLE 2.4 APPLICATION OF LEAST SQUARES TO DEVELOP A COST MODEL FOR THE COST OF HEAT EXCHANGERS

In the introduction we mentioned that it is sometimes necessary to develop a model for the objective function using cost data. Curve fitting of the costs of fabrication of heat exchangers can be used to predict the cost of a new exchanger of the same class with different design variables. Let the cost be expressed as a linear equation

$$C = \beta_1 + \beta_2 N + \beta_3 A$$

where β_1 , β_2 , and β_3 are constants

N = number of tubes

A = shell surface area

Estimate the values of the constants β_1 , β_2 , and β_3 from the data in Table E2.4. The regressors are $x_1 = 1$, $x_2 = N$, and $x_3 = A$.

Solution. The matrices to be used in calculating $\hat{\boldsymbol{\beta}}$ are as follows (each data set is weighted equally):

$$\mathbf{x} = \begin{bmatrix} 1 & 120 & 550 \\ 1 & 130 & 600 \\ 1 & 108 & 520 \\ 1 & 110 & 420 \\ 1 & 84 & 400 \\ 1 & 90 & 300 \\ 1 & 80 & 230 \\ 1 & 55 & 120 \\ 1 & 64 & 190 \\ 1 & 50 & 100 \end{bmatrix}$$

TABLE E2.4
Labor cost data for mild-steel
floating-head exchangers
(0–500 psig) working pressure

Labor cost (\$)	Area (A)	Number of tubes (N)
310	120	550
300	130	600
275	108	520
250	110	420
220	84	400
200	90	300
190	80	230
150	55	120
140	64	190
100	50	100

Source: Shahbenderian, 1961.

$$(\mathbf{x}^T\mathbf{x}) = \begin{bmatrix} 10 & 891 & 3,430 \\ 891 & 86,241 & 349,120 \\ 3,430 & 349,120 & 1,472,700 \end{bmatrix}$$

$$(\mathbf{x}^T\mathbf{Y}) = \begin{bmatrix} 2,135 \\ 207,290 \\ 844,800 \end{bmatrix}$$

Equation (2.9) gives the best estimates of β_1 , β_2 , and β_3 :

$$\hat{\beta}_1 = 38.177$$

$$\hat{\beta}_2 = 1.164$$

$$\hat{\beta}_3 = 0.209$$

Check to see if these coefficients yield a reasonable fit to the data in Table E2.4.

EXAMPLE 2.5 APPLICATION OF LEAST SQUARES IN YIELD CORRELATION

Ten data points were taken in an experiment in which the independent variable x is the mole percentage of a reactant and the dependent variable y is the yield (in percent):

x	y
20	73
20	78
30	85
40	90
40	91
50	87
50	86
50	91
60	75
70	65

Fit a quadratic model with these data and determine the value of x that maximizes the yield.

Solution. The quadratic model is $y = \beta_1 + \beta_2x + \beta_3x^2$. The estimated coefficients computed using Excel are

$$\hat{\beta}_1 = 35.66$$

$$\hat{\beta}_2 = 2.63$$

$$\hat{\beta}_3 = -0.032$$

The predicted optimum can be formed by differentiating

$$\hat{Y} = \hat{\beta}_1 + \hat{\beta}_2x + \hat{\beta}_3x^2$$

with respect to x and setting the derivative to zero to get

$$x^{\text{opt}} = \frac{-\hat{\beta}_2}{2\hat{\beta}_3} = 41.09$$

The predicted yield \hat{Y} at the optimum is 88.8.

Certain assumptions underly least squares computations such as the independence of the unobservable errors ε_i , a constant error variance, and lack of error in the x 's (Draper and Smith, 1998). If the model represents the data adequately, the residuals should possess characteristics that agree with these basic assumptions. The analysis of residuals is thus a way of checking that one or more of the assumptions underlying least squares optimization is not violated. For example, if the model fits well, the residuals should be randomly distributed about the value of y predicted by the model. Systematic departures from randomness indicate that the model is unsatisfactory; examination of the patterns formed by the residuals can provide clues about how the model can be improved (Box and Hill, 1967; Draper and Hunter, 1967).

Examinations of plots of the residuals versus \hat{Y}_i or x_i , or a plot of the frequency of the residuals versus the magnitude of the residuals, have been suggested as

numerical or graphical aids to assist in the analysis of residuals. A study of the signs of the residuals (+ or -) and sums of signs can be used. Residual analysis should include

1. Detection of an outlier (an extreme observation).
2. Detection of a trend in the residuals.
3. Detection of an abrupt shift in the level of the experiment (sequential observations).
4. Detection of changes in the error variance (usually assumed to be constant).
5. Examination to ascertain if the residuals are represented by a normal distribution (so that statistical tests can be applied).

When using residuals to determine the adequacy of a model, keep in mind that as more independent variables are added to the model, the residuals may become less informative. Each residual is, in effect, a weighted average of the ε_i 's; as more unnecessary x_i 's are added to a model, the residuals become more like one another, reflecting an indiscriminate average of all the ε 's instead of primarily representing one ε_i . In carrying out the analysis of residuals, you will quickly discover that a graphical presentation of the residuals materially assists in the diagnosis because one aberration, such as a single extreme value, can simultaneously affect several of the numerical tests.

Nonlinear least squares

If a model is nonlinear with respect to the model parameters, then nonlinear least squares rather than linear least squares has to be used to estimate the model coefficients. For example, suppose that experimental data is to be fit by a reaction rate expression of the form $r_A = kC_A^n$. Here r_A is the reaction rate of component A, C_A is the reactant concentration, and k and n are model parameters. This model is *linear* with respect to rate constant k but is *nonlinear* with respect to reaction order n . A general nonlinear model can be written as

$$y = f(x_1, x_2, x_3, \dots, \beta_1, \beta_2, \beta_3 \dots) \quad (2.10)$$

where y = the model output

x_j 's = model inputs

β_j 's = the parameters to be estimated

We still can define a sum-of-squares error criterion (to be minimized) by selecting the parameter set β_j so as to

$$\min_{\beta_j} \sum_{i=1}^n (Y_i - \hat{Y}_i)^2 \quad (2.11)$$

where Y_i = the i th output measurement

\hat{Y}_i = model prediction corresponding to the i th data point

The estimated coefficients listed for model 2 in Example 2.2 were obtained using nonlinear least squares (Bates and Watts, 1988).

As another example, consider the problem of estimating the gain K and time constants τ_i for first-order and second-order dynamic models based on a measured unit step response of the process $y(t)$. The models for the step response of these two processes are, respectively (Seborg et al., 1989),

$$y(t) = K(1 - e^{-t/\tau_1}) \quad (2.12)$$

$$y(t) = K \left(1 - \frac{\tau_1 e^{-t/\tau_1} - \tau_2 e^{-t/\tau_2}}{\tau_1 - \tau_2} \right) \quad (2.13)$$

where t = the independent variable (time)

y = the dependent variable

Although K appears linearly in both response equations, τ_1 in (2.12) and τ_1 and τ_2 in (2.13) appear nonlinearly, so that nonlinear least squares must be used to estimate their values. The specific details of how to carry out the computations will be deferred until we take up numerical methods of unconstrained optimization in Chapter 6.

2.4 FACTORIAL EXPERIMENTAL DESIGNS

Because variables in models are often highly correlated, when experimental data are collected, the $\mathbf{x}^T \mathbf{x}$ matrix in Equation 2.9 can be badly conditioned (see Appendix A), and thus the estimates of the values of the coefficients in a model can have considerable associated uncertainty. The method of factorial experimental design forces the data to be orthogonal and avoids this problem. This method allows you to determine the relative importance of each input variable and thus to develop a parsimonious model, one that includes only the most important variables and effects. Factorial experiments also represent efficient experimentation. You systematically plan and conduct experiments in which all of the variables are changed simultaneously rather than one at a time, thus reducing the number of experiments needed.

Because of the orthogonality property of factorial design, statistical tests are effective in discriminating among the effects of natural variations in raw materials, replicated unit operations (e.g., equipment in parallel), different operators, different batches, and other environmental factors. A proper orthogonal design matrix for collecting data provides independent estimates of the sums of squares for each variable as well as combinations of variables. Also the estimates of the coefficients have a lower variance than can be obtained with a nonorthogonal experimental design (Montgomery, 1997; Box et al., 1978). That is, you can have more confidence in the values calculated for β_i than would occur with a nonorthogonal design.

TABLE 2.1
Orthogonal experimental design

Experiment number	Response y	Scaled (coded) values of the independent variables	
		z_1	z_2
1	Y_1	-1	-1
2	Y_2	1	-1
3	Y_3	-1	1
4	Y_4	1	1
5	Y_5	0	0

From a practical standpoint, the user of the model must decide which input variables should be studied because this will determine the number of tests that must be carried out (Drain, 1997). In a standard factorial design, 2^n tests are required, where n is the number of input variables to be studied. You must also decide how much each input variable should be changed from its nominal value, taking into account the sensitivity of the process response to a change in a given input variable, as well as the typical operating range of the process. The determination of the region of experimentation requires process knowledge. The experimental range should be chosen so that the resulting measurements of the response do not involve errors in the sensors that are greater than typical noise levels.

Suppose you want to fit the linear model $y = \beta_1 + \beta_2 z_1 + \beta_3 z_2$, where z_1 and z_2 are the independent variables. Let the values of z_1 and z_2 in the experiment be deliberately chosen by an *experimental orthogonal design* like that shown in Table 2.1.

The values of the coded independent variables correspond to the four corners of a square in the z_1 and z_2 space. The summations in Equation (2.5) simplify in this case ($x_1 = 1$, $x_2 = z_1$, $x_3 = z_2$):

$$\sum_{i=1}^5 x_{i1}x_{i2} = \sum_{i=1}^5 z_{1i} = 0 \quad \sum_{i=1}^5 x_{i1}x_{i3} = \sum_{i=1}^5 z_{2i} = 0 \quad \sum_{i=1}^5 x_{i2}x_{i3} = \sum_{i=1}^5 z_{1i}z_{2i} = 0$$

$$\sum_{i=1}^5 x_{i1}x_{i1} = 5 \quad \sum_{i=1}^5 x_{i2}x_{i2} = \sum_{i=1}^5 z_{1i}^2 = 4 \quad \sum_{i=1}^5 x_{i3}x_{i3} = \sum_{i=1}^5 z_{2i}^2 = 4$$

For the experimental design in Table 2.1,

$$\mathbf{x}^T \mathbf{x} = \begin{bmatrix} 5 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 4 \end{bmatrix}$$

$$\mathbf{x}^T \mathbf{Y} = \begin{bmatrix} y_1 + y_2 + y_3 + y_4 + y_5 \\ -y_1 + y_2 - y_3 + y_4 \\ -y_1 - y_2 + y_3 + y_4 \end{bmatrix}$$

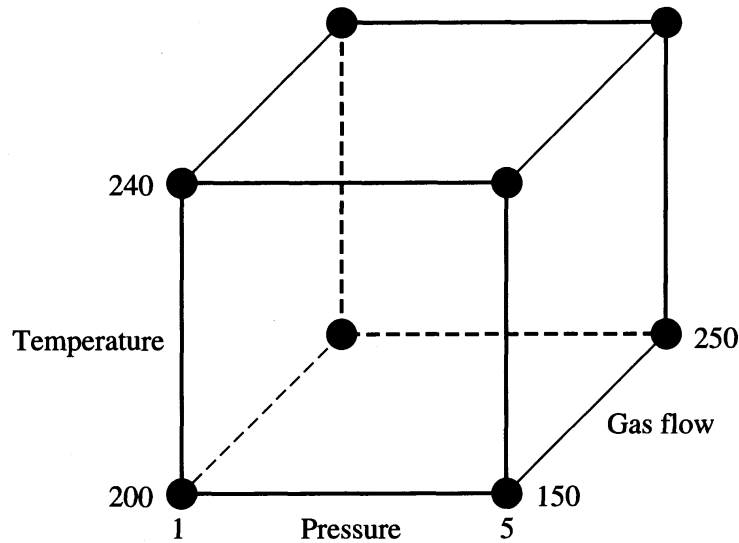


FIGURE E2.6
Orthogonal design for the variables temperature, pressure, and flowrate.

It is quite easy to solve Equation (2.9) now because these expressions are *uncoupled*; the inverse of $\mathbf{x}^T\mathbf{x}$ for Equation (2.13) can be obtained by merely taking the reciprocal of the diagonal elements.

EXAMPLE 2.6 IDENTIFICATION OF IMPORTANT VARIABLES BY EXPERIMENTATION USING AN ORTHOGONAL FACTORIAL DESIGN

Assume a reactor is operating at the reference state of 220°C, 3 atm pressure, and a gas flow rate of 200 kg/h. We can set up an orthogonal factorial design to model this process with a linear model $Y = \beta_1 + \beta_2x_2 + \beta_3x_3 + \beta_4x_4$ so that the coded values of the x_i are 1, -1, and 0. Examine Figure E2.6. Suppose we select the changes in the operating conditions of $\pm 20^\circ\text{C}$ for the temperature, ± 2 atm for the pressure, and ± 50 kg/h for flowrates. Let $x_1 = 1$; then x_2 , x_3 , and x_4 , the coded variables, are calculated in terms of the proposed operating conditions as follows:

$$x_2 = \frac{t(^{\circ}\text{C}) - 220}{20}$$

$$x_3 = \frac{p(\text{atm}) - 3}{2}$$

$$x_4 = \frac{m(\text{kg/h}) - 200}{50}$$

Based on the design the following data are collected:

Y (yield)	x_2	x_3	x_4
20.500	-1	-1	-1
60.141	1	-1	-1
58.890	-1	1	-1
67.712	1	1	-1
22.211	-1	-1	1
61.541	1	-1	1
59.902	-1	1	1
69.104	1	1	1
77.870	0	0	0
78.933	0	0	0
70.100	0	0	0

The extra data at the (0, 0) point are used to obtain a measure of the error involved in the experiment.

Solution. The matrices involved are

$$\mathbf{x}^T \mathbf{x} = \begin{bmatrix} 11 & 0 & 0 & 0 \\ 0 & 8 & 0 & 0 \\ 0 & 0 & 8 & 0 \\ 0 & 0 & 0 & 8 \end{bmatrix} \quad (\mathbf{x}^T \mathbf{x})^{-1} = \begin{bmatrix} 0.091 & 0 & 0 & 0 \\ 0 & 0.125 & 0 & 0 \\ 0 & 0 & 0.125 & 0 \\ 0 & 0 & 0 & 0.125 \end{bmatrix}$$

$$\mathbf{x}^T \mathbf{Y} = \begin{bmatrix} 646.9 \\ 96.99 \\ 91.21 \\ 5.51 \end{bmatrix}$$

With these matrices you can compute the estimates of $\hat{\beta}_i$ by solving Equation (2.9), yielding

$$\hat{Y} = 58.810 + 12.124x_2 + 11.402x_3 + 0.689x_4$$

In terms of the original variables

$$\begin{aligned} \hat{Y} &= 58.810 + 12.124 \left(\frac{t(^{\circ}\text{C}) - 220}{20} \right) + 11.402 \left(\frac{p(\text{atm}) - 3}{2} \right) \\ &\quad + 0.689 \left(\frac{m(\text{kg/h}) - 200}{50} \right) \\ &= 58.810 + 0.6062(t - 220) + 5.701(p - 3) + 0.0138(m - 200) \end{aligned}$$

It is clear from the size of the estimated coefficients that mass flowrate changes have a much smaller influence on the yield and thus, for practical purposes, could be eliminated as an important independent variable.

If the independent variables are orthogonal, deciding whether to add or delete variables or functions of variables in models is straightforward using stepwise least squares (regression), a feature available on many software packages. Stepwise regression consists of sequentially adding (or deleting) a variable (or function) of variables to a proposed model and then testing at each stage to see if the added (or deleted) variable is significant. The procedure is only effective when the independent variables are essentially orthogonal. The coupling of orthogonal experimental design with optimization of operating conditions has been called “evolutionary operation” by which the best operating conditions are determined by successive experiments (Box and Draper, 1969; Biles and Swain, 1980).

2.5 DEGREES OF FREEDOM

In Section 1.5 we briefly discussed the relationships of equality and inequality constraints in the context of independent and dependent variables. Normally in design and control calculations, it is important to eliminate redundant information and equations before any calculations are performed. Modern multivariable optimization software, however, does not require that the user clearly identify independent, dependent, or superfluous variables, or active or redundant constraints. If the number of independent equations is larger than the number of decision variables, the software informs you that no solution exists because the problem is overspecified. Current codes have incorporated diagnostic tools that permit the user to include all possible variables and constraints in the original problem formulation so that you do not necessarily have to eliminate constraints and variables prior to using the software. Keep in mind, however, that the smaller the dimensionality of the problem introduced into the software, the less time it takes to solve the problem.

The degrees of freedom in a model is the number of variables that can be specified independently and is defined as follows:

$$N_F = N_v - N_E \quad (2.14)$$

where N_F = degrees of freedom

N_v = total number of variables involved in the problem

N_E = number of independent equations (including specifications)

A degrees-of-freedom analysis separates modeling problems into three categories:

1. $N_F = 0$: *The problem is exactly determined.* If $N_F = 0$, then the number of independent equations is equal to the number of process variables and the set of equations may have a unique solution, in which case the problem is not an optimization problem. For a set of linear independent equations, a unique solution exists. If the equations are nonlinear, there may be no real solution or there may be multiple solutions.

2. $N_F > 0$: *The problem is underdetermined.* If $N_F > 0$, then more process variables exist in the problem than independent equations. The process model is said to be underdetermined, so at least one variable can be optimized. For linear models, the rank of the matrix formed by the coefficients indicates the number of independent equations (see Appendix A).
3. $N_F < 0$: *The problem is overdetermined.* If $N_F < 0$, fewer process variables exist in the problem than independent equations, and consequently the set of equations has no solutions. The process model is said to be overdetermined, and least squares optimization or some similar criterion can be used to obtain values of the unknown variables as described in Section 2.5.

EXAMPLE 2.7 MODEL FOR A SEPARATION TRAIN

Figure E2.7 shows the process flow chart for a series of two distillation columns, with mass flows and splits defined by x_1, x_2, \dots, x_5 . Write the material balances, and show that the process model comprises two independent variables and three degrees of freedom.

Solution. The balances for columns 1 and 2 are shown below:

$$\text{Column 1} \quad x_1 = x_2 + x_3 \quad \text{or} \quad x_1 - x_2 - x_3 = 0 \quad (a)$$

$$x_2 = .40x_1 \quad \text{or} \quad x_2 - 0.4x_1 = 0 \quad (b)$$

$$x_3 = .60x_1 \quad \text{or} \quad x_3 - 0.6x_1 = 0 \quad (c)$$

There are three equations and three unknowns.

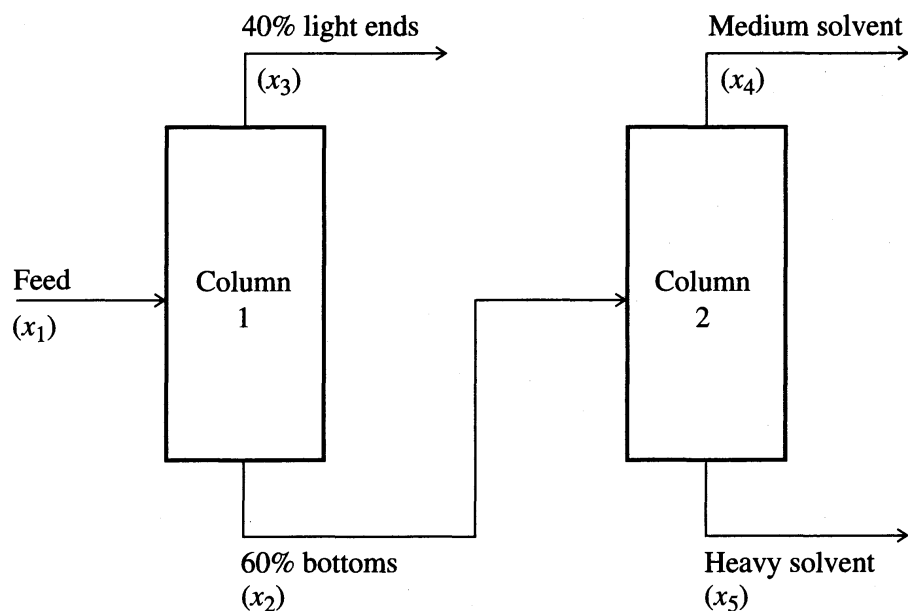


FIGURE E2.7
Train of distillation columns.

The coefficient matrix is

		Variables		
		x_1	x_2	x_3
Equations	(a)	1	-1	-1
	(b)	-0.4	1	0
	(c)	-0.6	0	1

The three equations are not independent. The rank of the coefficient matrix is 2, hence there are only two independent variables, and column 1 involves 1 degree of freedom.

$$\text{Column 2} \quad x_2 = x_4 + x_5 \quad \text{or} \quad x_2 - x_4 - x_5 = 0 \quad (d)$$

There is one equation and three unknowns, so there are two degrees of freedom. Overall there are four equations (a), (b), (c), (d) and five variables. The coefficient matrix is

	x_1	x_2	x_3	x_4	x_5
(a)	1	-1	-1	0	0
(b)	-0.4	1	0	0	0
(c)	-0.6	0	1	0	0
(d)	0	1	0	-1	-1

Because the rank of the coefficient matrix is three, there are only three independent equations, so Equation (2.14) indicates that there are two degrees of freedom. You can reduce the dimensionality of the set of material balances by substitution of one equation into another and eliminating both variables and equations.

In some problems it is advantageous to eliminate obvious dependent variables to reduce the number of equations that must be included as constraints. You can eliminate linear constraints via direct substitution, leaving only the nonlinear constraints, but the resulting equations may be too complex for this procedure to have merit. The following example illustrates a pipe flow problem in which substitution leads to one independent variable.

EXAMPLE 2.8 ANALYSIS OF PIPE FLOW

Suppose you want to design a hydrocarbon piping system in a plant between two points with no change in elevation and want to select the optimum pipe diameter that minimizes the combination of pipe capital costs and pump operating costs. Prepare a model that can be used to carry out the optimization. Identify the independent and dependent variables that affect the optimum operating conditions. Assume the fluid properties (μ , ρ) are known and constant, and the value of the pipe length (L) and mass flowrate (m) are specified. In your analysis use the following process variables: pipe diameter (D), fluid velocity (v), pressure drop (Δp), friction factor (f).

Solution. Intuitively one expects that an optimum diameter can be found to minimize the total costs. It is clear that the four process variables are related and not indepen-

dent, but we need to examine in an organized way how the equality constraints (models) affect the degrees of freedom.

List the equality constraints:

1. Mechanical energy balance, assuming no losses in fittings, no change in elevation, and so on.

$$\Delta p = \frac{2f\rho v^2 L}{D} \quad (a)$$

2. Equation of continuity, based on plug flow under turbulent conditions.

$$m = \left(\frac{\rho \pi D^2}{4} \right) v \quad (b)$$

3. A correlation relating the friction factor with the Reynolds number (Re).

$$f = f(\text{Re}) = f\left(\frac{Dv\rho}{\mu}\right)$$

The friction factor plot is available in many handbooks, so that given a value of Re, one can find the corresponding value of f . In the context of numerical optimization, however, using a graph is a cumbersome procedure. Because all of the constraints should be expressed as mathematical relations, we select the Blasius correlation for a smooth pipe (Bird et al., 1964):

$$f = 0.046 \text{Re}^{-0.2} = \frac{0.046\mu^{0.2}}{D^{0.2}v^{0.2}\rho^{0.2}} \quad (c)$$

The model involves four variables and three independent nonlinear algebraic equations, hence one degree of freedom exists. The equality constraints can be manipulated using direct substitution to eliminate all variables except one, say the diameter, which would then represent the independent variables. The other three variables would be dependent. Of course, we could select the velocity as the single independent variable of any of the four variables. See Example 13.1 for use of this model in an optimization problem.

2.6 EXAMPLES OF INEQUALITY AND EQUALITY CONSTRAINTS IN MODELS

As mentioned in Chapter 1, the occurrence of *linear inequality constraints* in industrial processes is quite common. Inequality constraints do not affect the count of the degrees of freedom unless they become active constraints. Examples of such constraints follow:

1. Production limitations arise because of equipment throughput restrictions, storage limitations, or market constraints (no additional product can be sold beyond some specific level).
2. Raw material limitations occur because of limitations in feedstock supplies; these supplies often are determined by production levels of other plants within the same company.
3. Safety or operability restrictions exist because of limitations on allowable operating temperatures, pressures, and flowrates.
4. Physical property specifications on products must be considered. In refineries the vapor pressure or octane level of fuel products must satisfy some specification. For blends of various products, you usually assume that a composite property can be calculated through the averaging of pure component physical properties. For N components with physical property values V_i and volume fraction y_i , the average property \bar{V} is

$$\bar{V} = \sum_{i=1}^N V_i y_i$$

EXAMPLE 2.9 FORMULATION OF A LINEAR INEQUALITY CONSTRAINT FOR BLENDING

Suppose three intermediates (light naphtha, heavy naphtha, and “catalytic” oil) made in a refinery are to be blended to produce an aviation fuel. The octane number of the fuel must be at least 95. The octane numbers for the three intermediates are shown in the table.

	Amount blended (barrels/day)	Octane number
Light naphtha	x_1	92
Heavy naphtha	x_2	86
Catalytic oil	x_3	97

Write an inequality constraint for the octane number of the aviation fuel, assuming a linear mixing rule.

Solution. Assume the material balance can be based on conservation of volume (as well as mass). The production rate of aviation gas is $x_4 = x_1 + x_2 + x_3$. The volume-average octane number of the gasoline can be computed as

$$\frac{x_1}{x_1 + x_2 + x_3} (92) + \frac{x_2}{x_1 + x_2 + x_3} (86) + \frac{x_3}{x_1 + x_2 + x_3} (97) \geq 95 \quad (a)$$

Multiplying Equation (a) by $(x_1 + x_2 + x_3)$ and rearranging, we get

$$-3x_1 - 9x_2 + 2x_3 \geq 0 \quad (b)$$

This constraint ensures that the octane number specification is satisfied. Note that Equation (b) is linear.

EXAMPLE 2.10 LINEAR MATERIAL BALANCE MODELS

In many cases in which optimization is applied, you need to determine the allocation of material flows to a set of processes in order to maximize profits. Consider the process diagram in Figure E2.10.

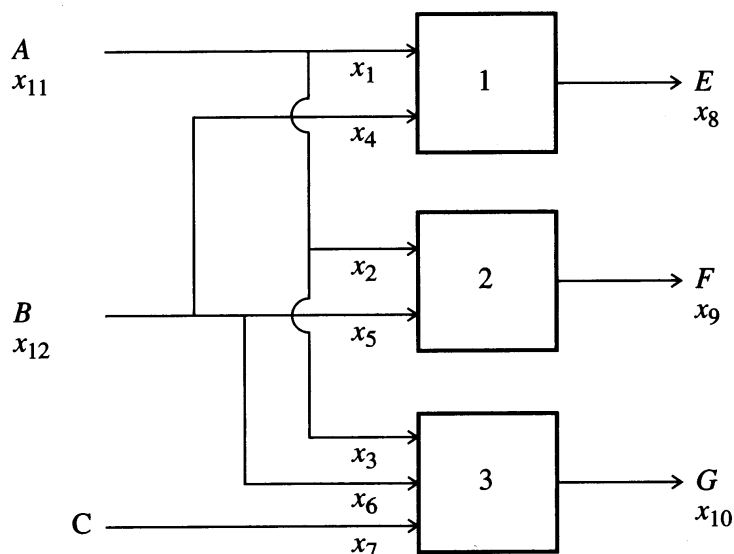


FIGURE E2.10
Flow diagram for a multiproduct plant.

Each product (E , F , G) requires different (stoichiometric) amounts of reactants according to the following mass balances:

Product	Reactants (1-kg product)
E	$\frac{2}{3}$ kg A, $\frac{1}{3}$ kg B
F	$\frac{2}{3}$ kg A, $\frac{1}{3}$ kg B
G	$\frac{1}{2}$ kg A, $\frac{1}{6}$ kg B, $\frac{1}{3}$ kg C

Prepare a model of the process using the mass balance equations.

Solution. Twelve mass flow variables can be defined for this process. Let x_1, x_2, x_3 be the mass input flows of A to each process. Similarly let x_4, x_5, x_6 , and x_7 be the individual reactant flows of B and C, and define x_8, x_9 , and x_{10} as the three mass product flows (E, F, G). Let x_{11} and x_{12} be the total amounts of A and B used as reactants (C is the same as x_7). Thus, we have a total of 12 variables.

The linear mass balance constraints that represent the process are:

$$A = x_{11} = x_1 + x_2 + x_3 \quad (a)$$

$$B = x_{12} = x_4 + x_5 + x_6 \quad (b)$$

$$x_1 = 0.667x_8 \quad (c)$$

$$x_2 = 0.667x_9 \quad (d)$$

$$x_3 = 0.5x_{10} \quad (e)$$

$$x_4 = 0.333x_8 \quad (f)$$

$$x_5 = 0.333x_9 \quad (g)$$

$$x_6 = 0.167x_{10} \quad (h)$$

$$x_7 = 0.333x_{10} \quad (i)$$

With 12 variables and 9 independent linear equality constraints, 3 degrees of freedom exist that can be used to maximize profits. Note that we could have added an overall material balance, $x_{11} + x_{12} + x_7 = x_8 + x_9 + x_{10}$, but this would be a redundant equation since it can be derived by adding the material balances.

Other constraints can be specified in this problem. Suppose that the supply of A was limited to 40,000 kg/day, or

$$x_{11} \leq 40,000 \quad (j)$$

If this constraint is inactive, that is, the optimum value of x_{11} is less than 40,000 kg/day, then, in effect, there are still 3 degrees of freedom. If, however, the optimization procedure yields a value of $x_{11} = 40,000$ (the optimum lies on the constraint, such as shown in Figure 1.2), then inequality constraint *f* becomes an equality constraint, resulting in only 2 degrees of freedom that can be used for optimization. You should recognize that it is possible to add more inequality constraints, such as constraints on materials supplies, in the model, for example,

$$x_{12} \leq 30,000 \quad (k)$$

$$x_7 \leq 25,000 \quad (l)$$

These can also become “active” constraints if the optimum lies on the constraint boundary. Note that we can also place inequality constraints on production of *E*, *F*, and *G* in order to satisfy market demand or sales constraints

$$x_8 \geq 20,000 \quad (m)$$

$$x_9 \geq 25,000 \quad (n)$$

$$x_{10} \geq 30,000 \quad (o)$$

Now the analysis is much more complex, and it is clear that more potential equality constraints exist than variables if all of the inequality constraints become active. It is possible that optimization could lead to a situation where no degrees of freedom would be left—one set of the inequality constraints would be satisfied as equalities. This outcome means no variables remain to be optimized, and the optimal solution reached would be at the boundaries, a subset of the inequality constraints.

Other constraints that can be imposed in a realistic problem formulation include

1. Operating limitations (bottlenecks)—there could be a throughput limitation on reactants to one of the processes (e.g., available pressure head).
 2. Environmental limitations—there could be some additional undesirable by-products *H*, such as the production of toxic materials (not in the original product list given earlier), that could contribute to hazardous conditions.
-

You can see that the model for a realistic process can become extremely complex; what is important to remember is that steps 1 and 3 in Table 1.1 provide an organized framework for identifying all of the variables and formulating the objective function, equality constraints, and inequality constraints. After this is done, you need not eliminate redundant variables or equations. The computer software can usually handle redundant relations (but not inconsistent ones).

REFERENCES

- Bates, D. M.; and D. G. Watts. *Nonlinear Regression Analysis and Its Applications*. Wiley, New York (1988).
- Biles, W. E.; and J. J. Swain. *Optimization and Industrial Experimentation*. Wiley-Interscience, New York (1980).
- Bird, R. B.; W. E. Stewart; and E. N. Lightfoot. *Transport Phenomena*. Wiley, New York (1964).
- Box, G. E. P.; and J. W. Hill. "Discrimination Among Mechanistic Models." *Technometrics* **9**: 57 (1967)
- Box, G. E. P.; and N. R. Draper. *Evolutionary Operation*. Wiley, New York (1969).
- Box, G. E. P.; W. G. Hunter, and J. S. Hunter. *Statistics for Experimenters*. Wiley-Interscience, New York (1978).
- Deitz, D. "Modeling Furnace Performance." *Mech Eng* **119** (Dec), 16 (1997).
- Drain, D. *Handbook of Experimental Methods for Process Improvements*. Chapman and Hall, International Thomson Publishing, New York (1997).
- Draper, N. R.; and W. G. Hunter. "The Use of Prior Distribution in the Design of Experiments for Parameter Estimation in Nonlinear Situations." *Biometrika* **54**: 147 (1967).
- Draper, N. R.; and H. Smith. *Applied Regression Analysis*, 3rd ed. Wiley, New York (1998).
- Eykhoff, P. *System Identification*. Wiley-Interscience, New York (1974).
- Jackson, J. E. *A User's Guide to Principal Components*. Wiley, New York (1991).
- McAdams, W. H. *Heat Transmission*. McGraw-Hill, New York (1954).
- Montgomery, D. C. *Design and Analysis of Experiments*, 4th ed. Wiley, New York (1997).
- Seborg, D. E.; T. F. Edgar; and D. A. Mellichamp. *Process Dynamics and Control*. Wiley, New York (1989).
- Shahbenderian, A. P. "The Application of Statistics to Cost Estimating." *Br Chem Eng* (January) **6**: 16 (1961).

SUPPLEMENTARY REFERENCES

- Bendor, E. A. *An Introduction to Mathematical Modeling*. John Wiley, New York (1978).
- Bequette, B. W. *Process Dynamics: Modeling, Analysis, and Simulation*. Prentice-Hall, Englewood Cliffs, NJ (1998).
- Churchill, S. W. *The Interpretation and Use of Rate Data*. Scripta Publishing Company, Washington, D. C. (1974).
- Davis, M. E. *Numerical Methods and Modeling for Chemical Engineers*. John Wiley, New York (1983).
- Denn, M. M. *Process Modeling*. Pitman Publishing, Marshfield, MA (1986).
- Friedly, J. C. *Dynamic Behavior of Processes*. Prentice-Hall, Englewood Cliffs, New Jersey (1972).

- Luyben, W. L. *Process Modeling, Simulation, and Control for Chemical Engineers*. McGraw-Hill, New York (1990).
- Montgomery, D. C.; G. C. Runger; and N. F. Hubele. *Engineering Statistics*. Wiley, New York (1998).
- Ogunnaike, T.; and W. H. Ray. *Process Dynamics, Modeling and Control*. Oxford University Press, New York (1994).
- Rice, R. G.; and D. D. Duong. *Applied Mathematics and Modeling for Chemical Engineers*. Wiley, New York (1995).
- Seider, W. D.; J. D. Seader; and D. R. Lewin. *Process Design Principles*. Wiley, New York (1999).
- Seinfeld, J. H.; and L. Lapidus. *Process Modeling, Estimation, and Identification*. Prentice-Hall, Englewood Cliffs, New Jersey (1974).
- Wen, C. Y.; and L. T. Fan. *Models for Flow Systems and Chemical Reactors*. Marcel Dekker, New York (1975).

PROBLEMS

2.1 Classify the following models as linear or nonlinear

(a) Two-pipe heat exchanger (streams 1 and 2)

$$\frac{\partial T_1}{\partial t} + v \frac{\partial T_1}{\partial z} = \frac{2h_1}{S_1 \rho_1 C_{p1}} (T_2 - T_1)$$

$$\frac{\partial T_2}{\partial t} = \frac{2h_1}{\rho_2 C_{p2} S_2} (T_2 - T_1)$$

$$BC: T_1(t, 0) = a \quad IC: T_1(0, z) = 0$$

$$T_2(t, 0) = b \quad T_2(0, z) = T_0$$

where T = temperature C_p = heat capacity

t = time S = area factor

BC = boundary conditions IC = initial conditions

ρ = density

(b) Diffusion in a cylinder

$$\frac{\partial C}{\partial t} = D \left(\frac{\partial^2 C}{\partial r^2} + \frac{1}{r} \frac{\partial C}{\partial r} \right)$$

$$C(0, r) = C_0$$

$$\frac{\partial C(t, 0)}{\partial r} = 0$$

$$C(t, R) = C_0$$

where C = concentration r = radial direction
 t = time D = constant

2.2 Classify the following equations as linear or nonlinear (y = dependent variable; x, z = independent variables)

(a) $y_1^2 + y_2^2 = a^2$

(b) $v_x \frac{\partial v_y}{\partial x} = \mu \frac{\partial^2 v_y}{\partial z^2}$

2.3 Classify the models in Problems 2.1 and 2.2 as steady state or unsteady state.

2.4 Classify the models in Problems 2.1 and 2.2 as lumped or distributed.

2.5 What type of model would you use to represent the process shown in the figure? Lumped or distributed? Steady state or unsteady state? Linear or nonlinear?

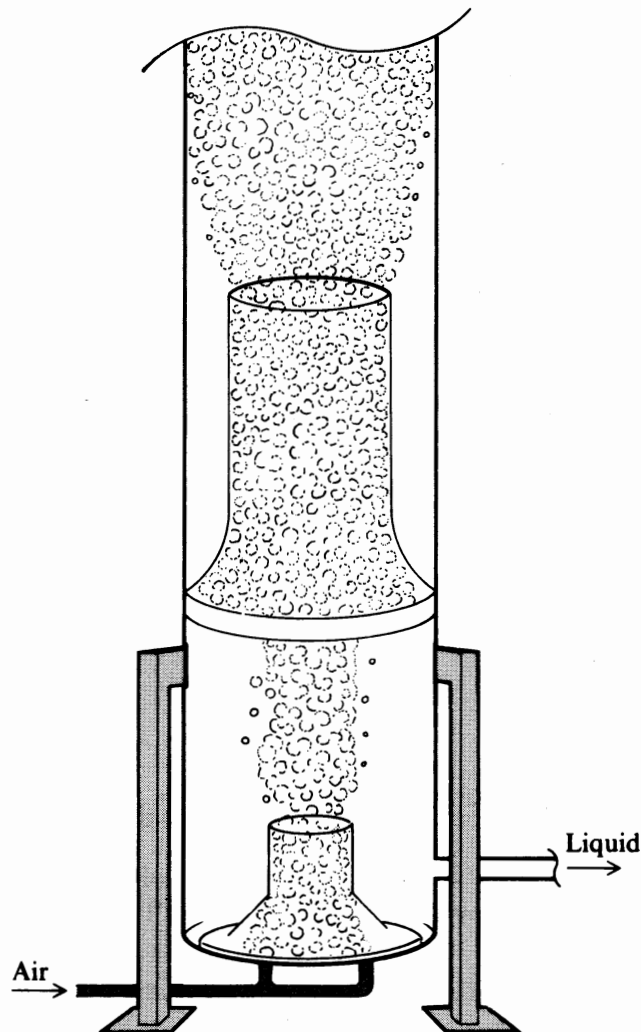


FIGURE P2.5

A wastewater treatment system uses five stacked venturi sections to ensure maximum oxygenation efficiency.

- 2.6 Determine the number of independent variables, the number of independent equations, and the number of degrees of freedom for the reboiler shown in the figure. What variables should be specified to make the solution of the material and energy balances determinate? (Q = heat transferred)

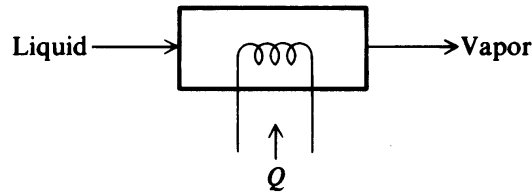


Figure P2.6

- 2.7 Determine the best functional relation to fit the following data sets:

(a)		(b)		(c)		(d)	
x_i	Y_i	x_i	Y_i	x_i	Y_i	x_i	Y_i
1	5	2	94.8	2	0.0245	0	8290
2	7	5	87.9	4	0.0370	20	8253
3	9	8	81.3	8	0.0570	40	8215
4	11	11	74.9	16	0.0855	60	8176
		14	68.7	32	0.1295	80	8136
		17	64.0	64	0.2000	100	8093
				128	0.3035		

- 2.8 The following data have been collected:

x_i	Y_i
10	1.0
20	1.26
30	1.86
40	3.31
50	7.08

Which of the following three models best represents the relationship between Y and x ?

$$y = e^{\alpha + \beta x}$$

$$y = e^{\alpha + \beta_1 x + \beta_2 x^2}$$

$$y = \alpha x^\beta$$

- 2.9 Given the following equilibrium data for the distribution of SO_3 in hexane, determine a suitable linear (in the parameters) empirical model to represent the data.

x_i pressure (psia)	Y_i weight fraction hexane
200	0.846
400	0.573
600	0.401
800	0.288
1000	0.209
1200	0.153
1400	0.111
1600	0.078

- 2.10** (a) Suppose that you wished to curve fit a set of data (shown in the table) with the equation

$$y = c_0 + c_1e^{3x} + c_2e^{-3x}$$

x_i	Y_i
0	1
1	2
2	2
3	1

Calculate c_0 , c_1 , and c_2 (show what summations need to be calculated). How do you find c_1 and c_2 if c_0 is set equal to zero?

- (b) If the desired equation were $y = a_1xe^{-a_2x}$, how could you use least-squares to find a_1 and a_2 ?

- 2.11** Fit the following data using the least squares method with the equation:

$$y = c_0 + c_1x$$

x_i	Y_i
0.5	0.6
1.0	1.4
2.1	2.0
3.4	3.6

Compare the results with a graphical (visual) estimate.

- 2.12** Fit the same data in Problem 2.11 using a quadratic fit. Repeat for a cubic model ($y = c_0 + c_1x + c_2x^2 + c_3x^3$). Plot the data and the curves.

- 2.13** You are asked to get the best estimates of the coefficients b_0 , b_1 , and c in the following model

$$y = b_0 + b_1e^{-cx}$$

given the following data.

Y_i	x_i
51.6	0.4
53.4	1.4
20.0	5.4
-4.2	19.5
-3.0	48.2
-4.8	95.9

Explain step by step how you would get the values of the coefficients.

- 2.14** Fit the following function for the density ρ as a function of concentration C , that is, determine the value of α in

$$\rho = \alpha + 1.33C$$

given the following measurements for ρ and C :

ρ (g/cm ³)	C (gmol/L)
3.31	1.01
4.69	1.97
5.92	3.11
7.35	4.00
8.67	4.95

- 2.15** (a) For the given data, fit a quadratic function of y versus x by estimating the values of all the coefficients.
 (b) Does this set of data constitute an orthogonal design?

y	6.4	5.6	6.0	7.5	6.5	8.3	7.7	11.7	10.3	17.6	18.0
x	1.0	1.0	1.0	2.0	2.0	3.0	3.0	4.0	4.0	5.0	5.0

- 2.16** Data obtained from a preset series of experiments was

Temperature, T (°F)	Pressure, p (atm)	Yield, Y (%)
160	1	4
160	1	5
160	7	10
160	7	11
200	1	24
200	1	26
200	7	35
200	7	38

Fit the linear model $\hat{Y} = b_0 + b_1x_1 + b_2x_2$ using the preceding table. Report the estimated coefficients b_0 , b_1 , and b_2 . Was the set of experiments a factorial design?

- 2.17** You are given data for Y versus x and asked to fit an empirical model of the form:

$$y = \alpha + \beta x$$

where β is a *known* value. Give an equation to calculate the best estimate of α .

- 2.18 A replicated two-level factorial experiment is carried out as follows (the dependent variables are yields):

Time (h)	Temperature (°C)	Yield (%)
1	240	24
5	240	42
1	280	3
5	280	19
1	240	24
5	240	46
1	280	5
5	280	21

Find the coefficients in a first-order model, $Y = \beta_0 + \beta_1x_1 + \beta_2x_2$. ($Y =$ yield, $x_1 =$ time, $x_2 =$ temperature.)

- 2.19 An experiment based on a hexagon design was carried out with four replications at the origin, producing the following data:

Factor levels			Design levels	
Yield (%)	Temperature (°C)	Time (h)	x_1	x_2
96.0	75	2.0	1.000	0
78.7	60	2.866	0.500	0.866
76.7	30	2.866	-0.500	0.866
54.6	15	2.0	-1.000	0
64.8	30	1.134	-0.500	-0.866
78.9	60	1.134	0.500	-0.866
97.4	45	2.0	0	0
90.5	45	2.0	0	0
93.0	45	2.0	0	0
86.3	45	2.0	0	0

Coding: $x_1 = \frac{\text{temperature} - 45}{30}$ $x_2 = \text{time} - 2$

Fit the full second-order (quadratic) model to the data.

- 2.20 A reactor converts an organic compound to product P by heating the material in the presence of an additive A . The additive can be injected into the reactor, and steam can be injected into a heating coil inside the reactor to provide heat. Some conversion can be obtained by heating without addition of A , and vice versa. In order to predict the yield of P , Y_p (lb mole product per lb mole feed), as a function of the mole fraction of A , X_A , and the steam addition S (in lb/lb mole feed), the following data were obtained.

Y_p	X_A	S
0.2	0.3	0
0.3	0	30
0.5	0	60

(a) Fit a linear model

$$Y_p = c_0 + c_1 X_A + c_2 S$$

that provides a least squares fit to the data.

(b) If we require that the model always must fit the point $Y_p = 0$ for $X_A = S = 0$, calculate c_0 , c_1 , and c_2 so that a least squares fit is obtained.

2.21 If you add a feed stream to the equilibrium stage shown in the figure, determine the number of degrees of freedom for a binary mixture (Q = heat transferred).

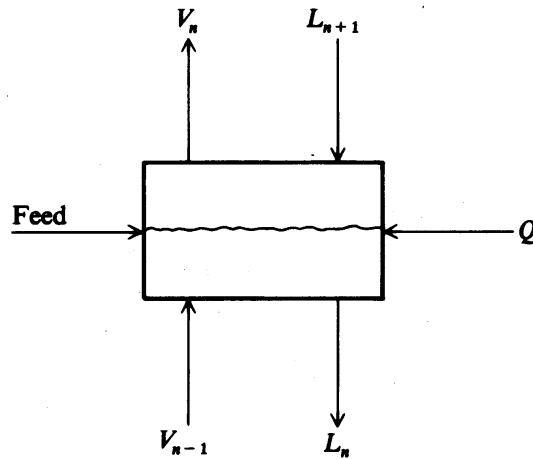


FIGURE P2.21

2.22 How many variables should be selected as independent variables for the furnace shown in the figure?

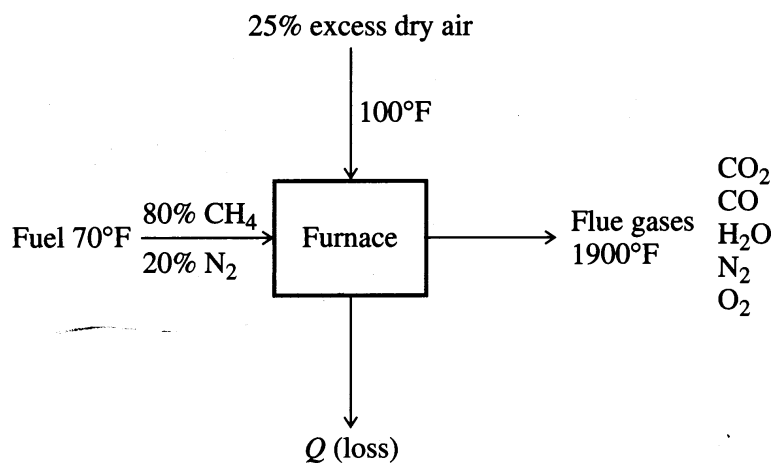


FIGURE P2.22

2.23 Determine the number of independent variables, the number of independent equations, and the number of degrees of freedom in the following process (A , B , and D are chemical species):

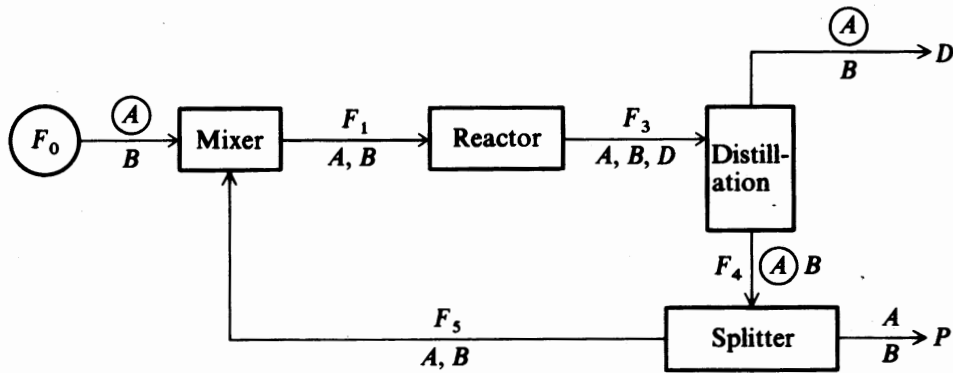


FIGURE P2.23

The encircled variables have known values. The reaction parameters in the reactor are known as the fraction split at the splitter between F_4 and F_5 . Each stream is a single phase.

2.24 A waste heat boiler (see Fig. P2.24) is to be designed for steady-state operation under the following specifications.

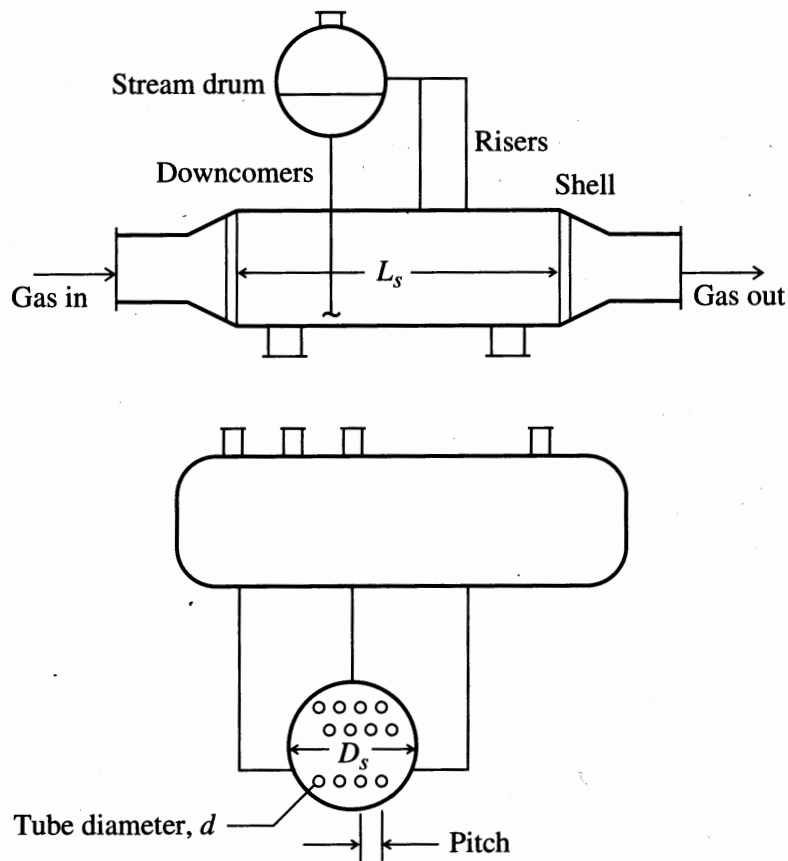


FIGURE P2.24

Total gas flow	25,000 kg/h
Gas composition	SO ₂ (9%), O ₂ (12%), N ₂ (79%)
Gas temperatures	in = 1200°C; out = 350°C
Stream pressure outside tubes	250 kPa
Gas properties	$C_p = 0.24 \text{ kcal/(g)(°C)}$ $\mu = 0.14 \text{ kg/(m)(h)}$ $k = 0.053 \text{ kcal/(m)(h)(°C)}$

Cost data are

Shell	\$2.50/kg
Tubes	\$150/m ²
Electricity	\$0.60/kWh
Interest rate	14%

Base the optimization on just the cost of the shell, tubes, and pumping costs for the gas. Ignore maintenance and repairs.

Formulate the optimization problem using only the following notation (as needed):

A	surface area of tubes, m ²
C_s	cost of shell, \$
C_t	cost of tubes, \$
C_{pi}	heat capacity of gas, kcal/(kg)(°C)
D	diameter of shell, m
d_o, d_i	tube outer and inner diameters, m
f	friction factor
g	acceleration due to gravity, m/s ²
h_i	gas side heat transfer coefficient inside the tubes, kcal/(m ²)(h)(°C)
i	interest rate, fraction
k	gas thermal conductivity, kcal/(m)(h)(°C)
L_s	length of shell, m
MW	molecular weight of gas
n	number of tubes
N	life of equipment, years
Q	duty of the boiler, kcal/h
T_1, T_2	gas temperature entering and leaving the boiler, °C
T	temperature in general
ρ_g	density of gas, kg/m ³
μ_g	viscosity of gas, kg/(m)(h)
V	gas velocity, m/s
W_g	gas flow, kg/h
W_s	weight of shell, tons
η	efficiency of blower
ΔP_g	gas pressure drop, kPa
Z	shell thickness, m

How many degrees of freedom are in the problem you formulated?

**FORMULATION OF THE OBJECTIVE
FUNCTION**

3.1 Economic Objective Functions 84
3.2 The Time Value of Money in Objective Functions 91
3.3 Measures of Profitability 100
References 104
Supplementary References 104
Problems 105

THE FORMULATION OF objective functions is one of the crucial steps in the application of optimization to a practical problem. As discussed in Chapter 1, you must be able to translate a verbal statement or concept of the desired objective into mathematical terms. In the chemical industries, the objective function often is expressed in units of currency (e.g., U.S. dollars) because the goal of the enterprise is to minimize costs or maximize profits subject to a variety of constraints. In other cases the problem to be solved is the maximization of the yield of a component in a reactor, or minimization of the use of utilities in a heat exchanger network, or minimization of the volume of a packed column, or minimizing the differences between a model and some data, and so on. Keep in mind that when formulating the mathematical statement of the objective, functions that are more complex or more nonlinear are more difficult to solve in optimization. Fortunately, modern optimization software has improved to the point that problems involving many highly nonlinear functions can be solved.

Although some problems involving multiple objective functions cannot be reduced to a single function with common units (e.g., minimize cost while simultaneously maximizing safety), in this book we will focus solely on scalar objective functions. Refer to Hurvich and Tsai (1993), Kamimura (1997), Rusnak et al. (1993), or Steur (1986) for treatment of multiple objective functions. You can, of course, combine two or more objective functions by trade-off, that is, by suitable weighting (refer to Chapter 8). Suppose you want to maintain the quality of a product in terms of two of its properties. One property is the deviation of the variable y_i (i designates the sample number) from the setpoint for the variable, y_{sp} . The other property is the variability of y_i from its mean \bar{y} (which during a transition may not be equal to y_{sp}). If you want to simultaneously use both criteria, you can minimize f :

$$f = w_1 \sum_i \left[y_{sp} - y_i \right]^2 + w_2 \sum_i \left[y_i - \bar{y} \right]^2 \quad (3.1)$$

where the w_i are weighting factors to be selected by engineering judgment. From this viewpoint, you can also view each term in the summations as being weighted equally.

This chapter includes a discussion of how to formulate objective functions involved in economic analysis, an explanation of the important concept of the time value of money, and an examination of the various ways of carrying out a profitability analysis. In Appendix B we cover, in more detail, ways of estimating the capital and operating costs in the process industries, components that are included in the objective function. For examples of objective functions other than economic ones, refer to the applications of optimization in Chapters 11 to 16.

3.1 ECONOMIC OBJECTIVE FUNCTIONS

The ability to understand and apply the concepts of cost analysis, profitability analysis, budgets, income-and-expense statements, and balance sheets are key skills that may be valuable. This section treats two major components of economic

objective functions: capital costs and operating costs. Economic decisions are made at various levels of detail. The more detail involved, the greater the expense of preparing an economic study. In engineering practice you may need to prepare preliminary cost estimates for projects ranging from a small piece of equipment or a new product to a major plant retrofit or design.

To introduce the involvement of these two types of costs in an objective function, we consider three simple examples: The first involves only operating costs and income, the second involves only capital costs, and the third involves both.

EXAMPLE 3.1 OPERATING PROFITS AS THE OBJECTIVE FUNCTION

Let us return to the chemical plant of Example 2.10 with three products (E, F, G) and three raw materials (A, B, C) in limited supply. Each of the three products is produced in a separate process (1, 2, 3); Figure E3.1 illustrates the process.

Process data

Process 1: $A + B \rightarrow E$

Process 2: $A + B \rightarrow F$

Process 3: $3A + 2B + C \rightarrow G$

Raw material	Maximum available (kg/day)	Cost (¢/kg)
A	40,000	1.5
B	30,000	2.0
C	25,000	2.5

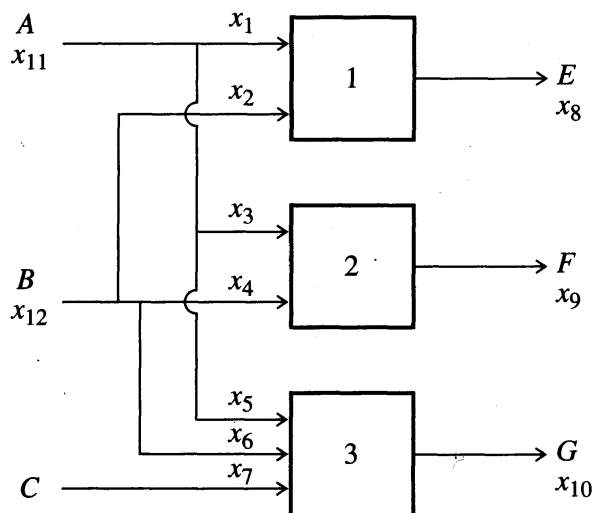


FIGURE E3.1
Flow diagram for a multiproduct plant.

Process	Product	Reactant requirements (kg/kg product)	Processing cost (product) (¢/kg)	Selling price (product) (¢/kg)
1	E	$\frac{2}{3}A, \frac{1}{3}B$	1.5	4.0
2	F	$\frac{2}{3}A, \frac{1}{3}B$	0.5	3.3
3	G	$\frac{1}{2}A, \frac{1}{6}B, \frac{1}{3}C$	1.0	3.8

(mass is conserved)

Formulate the objective function to maximize the total operating profit per day in the units of \$/day.

Solution The notation for the mass flow rates of reactants and products is the same as in Example 2.10.

The income in dollars per day from the plant is found from the selling prices ($0.04E + 0.033F + 0.038G$). The operating costs in dollars per day include

$$\text{Raw material costs: } 0.015A + 0.02B + 0.025C$$

$$\text{Processing costs: } 0.015E + 0.005F + 0.01G$$

$$\begin{aligned} \text{Total costs in dollars per day} &= 0.015A + 0.02B + 0.025C + 0.015E \\ &\quad + 0.005F + 0.01G \end{aligned}$$

The daily profit is found by subtracting daily operating costs from the daily income:

$$\begin{aligned} f(\mathbf{x}) &= 0.025E + 0.028F + 0.028G - 0.015A - 0.02B - 0.025C \\ &= 0.025x_8 + 0.028x_9 + 0.028x_{10} - 0.015x_{11} - 0.02x_{12} - 0.025x_7 \end{aligned}$$

Note that the six variables in the objective function are constrained through material balances, namely

$$x_{11} = 0.667x_8 + 0.667x_9 + 0.5x_{10}$$

$$x_{12} = 0.333x_8 + 0.333x_9 + 0.167x_{10}$$

$$x_7 = 0.333x_{10}$$

Also

$$0 \leq x_{11} \leq 40,000$$

$$0 \leq x_{12} \leq 30,000$$

$$0 \leq x_7 \leq 25,000$$

The optimization problem in this example comprises a linear objective function and linear constraints, hence linear programming is the best technique for solving it (refer to Chapter 7).

The next example treats a case in which only capital costs are to be optimized.

EXAMPLE 3.2 CAPITAL COSTS AS THE OBJECTIVE FUNCTION

Suppose you wanted to find the configuration that minimizes the capital costs of a cylindrical pressure vessel. To select the best dimensions (length L and diameter D) of the vessel, formulate a suitable objective function for the capital costs and find the optimal (L/D) that minimizes the cost function. Let the tank volume be V , which is fixed. Compare your result with the design rule-of-thumb used in practice, $(L/D)^{\text{opt}} = 3.0$.

Solution Let us begin with a simplified geometry for the tank based on the following assumptions:

1. Both ends are closed and flat.
2. The vessel walls (sides and ends) are of constant thickness t with density ρ , and the wall thickness is not a function of pressure.
3. The cost of fabrication and material is the same for both the sides and ends, and is S (dollars per unit weight).
4. There is no wasted material during fabrication due to the available width of metal plate.

The surface area of the tank using these assumptions is equal to

$$2\left(\frac{\pi D^2}{4}\right) + \pi DL = \frac{\pi D^2}{2} + \pi DL \quad (a)$$

(ends) (cylinder)

From assumptions 2 and 3, you might set up several different objective functions:

$$f_1 = \frac{\pi D^2}{2} + \pi DL \quad (\text{units of area}) \quad (b)$$

$$f_2 = \rho \left(\frac{\pi D^2}{2} + \pi DL \right) \cdot t \quad (\text{units of weight}) \quad (c)$$

$$f_3 = S \cdot \rho \cdot \left(\frac{\pi D^2}{2} + \pi DL \right) \cdot t \quad (\text{units of cost in dollars}) \quad (d)$$

Note that all of these objective functions differ from one another only by a multiplicative constant; this constant has no effect on the values of the independent variables at the optimum. For simplicity, we therefore use f_1 to determine the optimal values of D and L . Implicit in the problem statement is that a relation exists between volume and length, namely the constraint

$$V = \frac{\pi D^2}{4} \cdot L \quad (e)$$

Hence, the problem has only one independent variable.

Next use (e) to remove L from (b) to obtain the objective function

$$f_4 = \frac{\pi D^2}{2} + \frac{4V}{D} \quad (f)$$

Differentiation of f_4 with respect to D for constant V , equating the derivative to zero, and solving the resulting equation gives

$$D^{\text{opt}} = \left(\frac{4V}{\pi} \right)^{1/3} \quad (g)$$

This result implies that $f_4 \sim V^{2/3}$, a relationship close to the classical "six-tenths" rule used in cost estimating. From (e), $L^{\text{opt}} = (4V/\pi)^{1/3}$; this yields a rather surprising result, namely

$$\left(\frac{L}{D} \right)^{\text{opt}} = 1 \quad (h)$$

The $(L/D)^{\text{opt}}$ ratio is significantly different from the rule of thumb stated earlier in the example, namely, $L/D = 3$; this difference must be due to the assumptions (perhaps erroneous) regarding vessel geometry and fabrication costs.

Brummerstedt (1944) and Happel and Jordan (1975) discussed a somewhat more realistic formulation of the problem of optimizing a vessel size, making the following modifications in the original assumptions:

1. The ends of the vessel are 2:1 ellipsoidal heads, with an area for the two ends of $2(1.16D^2) = 2.32D^2$.
2. The cost of fabrication for the ends is higher than the sides; Happel and Jordan suggested a factor of 1.5.
3. The thickness t is a function of the vessel diameter, allowable steel stress, pressure rating of the vessel, and a corrosion allowance. For example, a design pressure of 250 psi and a corrosion allowance of $\frac{1}{8}$ in. give the following formula for t in inches (in which D is expressed in feet):

$$t = 0.0108D + 0.125 \quad (i)$$

The three preceding assumptions require that the objective function be expressed in dollars since area and weight are no longer directly proportional to cost

$$f_5 = \rho [\pi DLS + (1.5S)(2.32D^2)]t \quad (j)$$

The unit conversion of t from inches to feet does not affect the optimum (L/D) , nor do the values of ρ and S , which are multiplicative constants. The modified objective function, substituting Equation (i) in Equation (j), is therefore

$$f_6 = 0.0339D^2L + 0.435D^2 + 0.3927DL + 0.0376D^3 \quad (k)$$

The volume constraint is also different from the one previously used because of the dished heads:

$$V = \frac{\pi D^2}{4} \left(L + \frac{D}{3} \right) \quad (l)$$

Equation (l) can be solved for L and substituted into Equation (k). However, No analytical solution for D^{opt} by direct differentiation of the objective function is possible

now because the expression for f_6 , when L is eliminated, leads to a complicated polynomial equation for the objective function:

$$f_7 = 0.0432V + 0.5000 \frac{V}{D} + 0.3041D^2 + 0.0263D^3 \quad (m)$$

When f_7 is differentiated, a fourth-order polynomial in D results; no simple analytical solution is possible to obtain the optimum value of D . A numerical search is therefore better for obtaining D^{opt} and should be based on f_7 (rather than examining $df_7/dD = 0$). However, such a search will need to be performed for different values of V and the design pressure, parameters which are embedded in Equation (i). Recall that Equations (i) and (m) are based on a design pressure of 250 psi. Happel and Jordan (1975) presented the following solution for $(L/D)^{\text{opt}}$:

TABLE E3.2
Optimum (L/D)

Capacity (gal)	Design pressure (psi)		
	100	250	400
2,500	1.7	2.4	2.9
25,000	2.2	2.9	4.3

In Chapter 5 you will learn how to obtain such a solution. Note that for small capacities and low pressures, the optimum L/D approaches the ideal case; examine Equation (h) considered earlier. It is clear from Table E3.2 that the rule of thumb that $(L/D)^{\text{opt}} = 3$ can be in error by as much as ± 50 percent from the actual optimum. Also, the optimum does not take into account materials wasted during fabrication, a factor that could change the answer.

Next we consider an example in which both operating costs and capital costs are included in the objective function. The solution of this example requires that the two types of costs be put on some common basis, namely, dollars per year.

EXAMPLE 3.3 OPTIMUM THICKNESS OF INSULATION

In specifying the insulation thickness for a cylindrical vessel or pipe, it is necessary to consider both the costs of the insulation and the value of the energy saved by adding the insulation. In this example we determine the optimum thickness of insulation for a large pipe that contains a hot liquid. The insulation is added to reduce heat losses from the pipe. Next we develop an analytical expression for insulation thickness based on a mathematical model.

The rate of heat loss from a large insulated cylinder (see Figure E3.3), for which the insulation thickness is much smaller than the cylinder diameter and the inside heat transfer coefficient is very large, can be approximated by the formula

$$Q = \frac{A\Delta T}{x/k + 1/h_c} \quad (a)$$

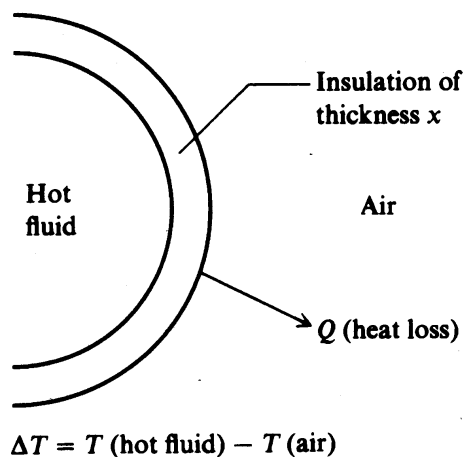


FIGURE E3.3
Heat loss from an insulated pipe

where ΔT = average temperature difference between pipe fluid and ambient surroundings, K

A = surface area of pipe, m^2

x = thickness of insulation, m

h_c = outside convective heat transfer coefficient, $\text{kJ}/(\text{h})(\text{m}^2)(\text{K})$

k = thermal conductivity of insulation, $\text{kJ}/(\text{h})(\text{m})(\text{K})$

Q = heat loss, kJ/h

All of the parameters on the right hand side of Equation (a) are fixed values except for x , the variable to be optimized. Assume the cost of installed insulation per unit area can be represented by the relation $C_0 + C_1x$, where C_0 and C_1 are constants (C_0 = fixed installation cost and C_1 = incremental cost per foot of thickness). The insulation has a lifetime of 5 years and must be replaced at that time. The funds to purchase and install the insulation can be borrowed from a bank and paid back in five annual installments. Let r be the fraction of the installed cost to be paid each year to the bank. The value of r selected depends on the interest rate of the funds borrowed and will be explained in Section 3.2.

Let the value of the heat lost from the pipe be H_t ($\$/10^6$ kJ). Let Y be the number of hours per year of operation. The problem is to

1. Formulate an objective function to maximize the savings in operating cost, savings expressed as the difference between the value of the heat conserved less the annualized cost of the insulation.
2. Obtain an analytical solution for x^* , the optimum.

Solution If operating costs are to be stated in terms of dollars per year, then the capital costs must be stated in the same units. Because the funds required for the insulation are to be paid back in equal installments over a period of 5 years, the payment per year is $r(C_0 + C_1x)A$. The energy savings due to insulation can be calculated from the difference between $Q(x = 0) = Q_0$, and Q :

$$Q_0 - Q = h_c \Delta T A - \frac{\Delta T A}{x/k + 1/h_c} \quad (b)$$

The objective function to be maximized is the present value of heat conserved in dollars less the annualized capital cost (also in dollars):

$$f = (Q_0 - Q) \left(\frac{\text{kJ}}{\text{h}} \right) \cdot Y \left(\frac{\text{h}}{\text{year}} \right) \cdot H_t \left(\frac{\text{dollars}}{\text{kJ}} \right) \frac{1}{r} (\text{year}) - (C_0 + C_1 x) A (\text{dollars}) \quad (c)$$

Substitute Equation (b) into (c), differentiate f with respect to x , and solve for the optimum ($df/dx = 0$):

$$x^* = k \left\{ \left(\frac{H_t Y \Delta T}{10^6 k C_1 r} \right)^{1/2} - \frac{1}{h_c} \right\} \quad (d)$$

Examine how x^* varies with the different parameters in (d), and confirm that the trends are physically meaningful. Note that the heat transfer area A does not appear in Equation (d). Why? Could you formulate f as a cost minimization problem, that is, the sum of the value of heat lost plus insulation cost? Does it change the result for x^* ? How do you use this result to select the correct commercial insulation size (see Example 1.1)?

Appendix B explains ways of estimating the capital and operating costs, leading to the coefficients in economic objective functions.

3.2 THE TIME VALUE OF MONEY IN OBJECTIVE FUNCTIONS

So far we have explained how to estimate capital and operating costs. In Example 3.3, we formulated an objective function for economic evaluation and discovered that although the revenues and operating costs occur in the future, most capital costs are incurred at the beginning of a project. How can these two classes of costs be evaluated fairly? The economic analysis of projects that incur income and expense over time should include the concept of the time value of money. This concept means that a unit of money (dollar, yen, euro, etc.) on hand now is worth more than the same unit of money in the future. Why? Because \$1000 invested today can earn additional dollars; in other words, the value of \$1000 received in the future will be less than the present value of \$1000.

For an example of the kinds of decisions that involve the time value of money, examine the advertisement in Figure 3.1. For which option do you receive the most value? Answers to this and similar questions sometimes may be quickly resolved using a calculator or computer without much thought. To understand the underlying assumptions and concepts behind the calculations, however, you need to account for cash flows in and out using the investment time line diagram for a project. Look at Figure 3.2.

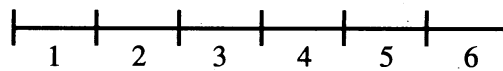
You Decide Which Option You Prefer If You Are The Winner Of The Sweepstakes:

Option 1	OR	Option 2	OR	Option 3
\$2,000,000 NOW. Payable immediately.		\$1,000,000 NOW. PLUS \$137,932 a year for 29 years.		\$167,000 a year for 30 years.

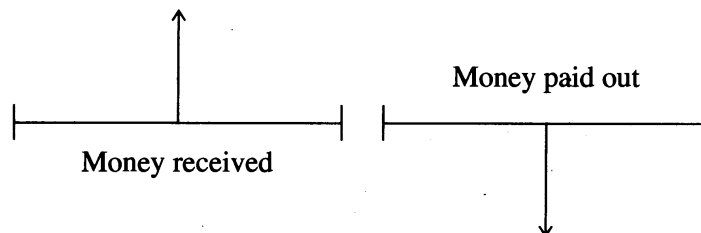
Tell us your choice. Read the instructions on the reverse to learn how you can activate your Grand Prize Option.

FIGURE 3.1

Options for potential sweepstakes winners. Which option provides the optimal value?

**FIGURE 3.2**

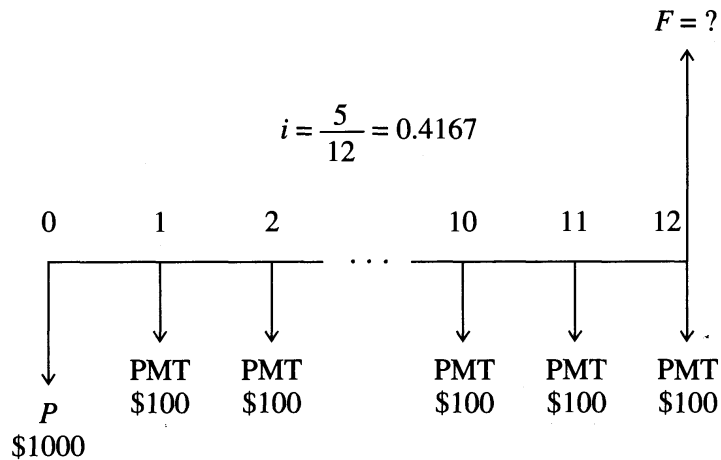
The time line with divisions corresponding to 6 time periods.

**FIGURE 3.3**

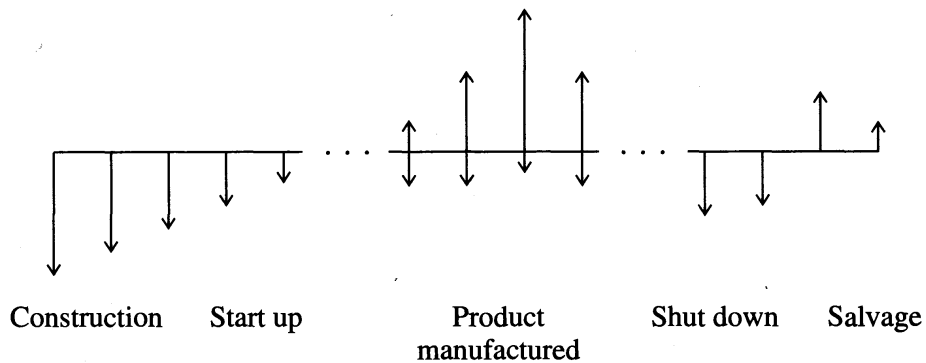
Representation of cash received and disbursed.

Figure 3.3 depicts money received (or income) with vertical arrows pointing upward; money paid out (or expenses) is depicted by vertical arrows pointing downward. With the aid of Figure 3.3 you can represent almost any complicated financial plan for a project. For example, suppose you deposit \$1000 now (the present value P) in a bank savings account that pays 5.00 percent annual interest compounded monthly, and in addition you plan to deposit \$100 per month at the end of each month for the next year. What will the future value F of your investments be at the end of the year? Figure 3.4 outlines the arrangement on the time line.

Note that cash flows corresponding to the accrual of interest are not represented by arrows in Figure 3.4. The interest rate per month is 0.4167, not 5.00 percent (the

**FIGURE 3.4**

The transactions for the example placed on the time line.

**FIGURE 3.5**

Cash flow transactions for a proposed plant placed on the time line.

annual interest rate). The number of compounding periods is $n = 12$. PMT is the periodic payment.

Figure 3.5 shows (using arrows only) some of the typical cash flows that might occur from the start to the end of a proposed plant. As the plant is built, the cash flows are negative, as is most likely the case during startup. Once in operation, the plant produces positive cash flows that diminish with time as markets change and competitors start up. Finally, the plant is closed, and eventually the equipment sold or scrapped.

It is easy to develop a general formula for investment growth for the case in which fractional interest i is compounded once per period (month, year). (*Note:* On most occasions we will cite i in percent, as is the common practice, even though in problem calculations i is treated as a fraction.) If P is the original investment (*present value*), then $P(1 + i)$ is the amount accumulated after one compounding period,

say 1 year. Using the same reasoning, the value of the investment in successive years for discrete interest payments is

$$t = 2 \text{ years} \quad F_2 = P(1 + i) + iP(1 + i) = P(1 + i)^2 \quad (3.2a)$$

$$t = 3 \text{ years} \quad F_3 = P(1 + i)^2 + iP(1 + i)^2 = P(1 + i)^3 \quad (3.2b)$$

$$t = n \text{ years} \quad F_n = P(1 + i)^n \quad (3.2c)$$

The symbol F_n is called the *future worth* of the investment after year n , that is, the future value of a current investment P based on a specific interest rate i .

Equation (3.2c) can be rearranged to give present value in terms of future value, that is, the present value of one future payment F at period n

$$P = \frac{F_n}{(1 + i)^n} \quad (3.3)$$

For continuous compounding Equation (3.2c) reduces to $F_n = Pe^{in}$. Refer to Garrett, Chapt. 5 (1989) for the derivation of this formula.

The following is a list of some useful extensions of Equation (3.3). Note that the factors involved in Equations (3.3)–(3.7) are F , P , i , and n , and given the values of any three, you can calculate the fourth. Software such as Microsoft Excel and hand calculators all contain programs to execute the calculations, many of which must be iterative.

1. Present value of a series of payments F_k (not necessarily equal) at periods $k = 1, \dots, n$ in the future:

$$P = \frac{F_1}{(1 + i)} + \frac{F_2}{(1 + i)^2} + \dots + \frac{F_{n-1}}{(1 + i)^{n-1}} + \frac{F_n}{(1 + i)^n} \quad (3.4)$$

$$= \sum_{k=1}^n \frac{F_k}{(1 + i)^k} \quad (3.4a)$$

2. Present value of a series of uniform future payments each of value 1 starting in period m and ending with period n :

$$\begin{aligned} P &= \sum_{k=m}^n \frac{1}{(1 + i)^k} = \left[-\left(\frac{1 + i}{i}\right) \left(\frac{1}{1 + i}\right)^k \right]_m^{n+1} = \frac{1}{i(1 + i)^{m-1}} - \frac{1}{i(1 + i)^n} \\ &= \frac{(1 + i)^{n-m+1} - 1}{i(1 + i)^n} \end{aligned}$$

If $m = 1$,

$$P = \sum_{k=1}^n \frac{1}{(1 + i)^k} = \frac{(1 + i)^n - 1}{i(1 + i)^n} \quad (3.5)$$

3. Future value of a series of (not necessarily equal) payments P_k :

$$F = \sum_{k=1}^n P_k (1 + i)^{n-k+1} \quad (3.6)$$

4. Future value of a series of uniform future payments each of value 1 starting in period m and ending in period n :

$$F = (1 + i)^n \sum_{k=m}^n \frac{1}{(1 + i)^k} = \frac{(1 + i)^{n-m+1} - 1}{i} \quad (3.7)$$

If $m = 1$ so that $k = 1$, the equivalent of Equation (3.7) is

$$F = (1 + i)^n \sum_{k=1}^n \frac{1}{(1 + i)^k} = \frac{(1 + i)^n - 1}{i}$$

The right-hand side of Equation (3.5) is known as the “capital recovery factor” or “present worth factor,” and the inverse of the right-hand side is known as the “repayment multiplier” r .

$$r = \frac{i(1 + i)^n}{(1 + i)^n - 1} \quad (3.8)$$

Tables of the repayment multiplier are listed in handbooks and some textbooks. Table 3.1 gives r over some limited ranges as a function of n and i .

TABLE 3.1

Values for the fraction $r = \frac{i(1 + i)^n}{(1 + i)^n - 1}$

Interest rate										
n	$i \rightarrow 1$	2	4	6	8	10	12	14	16	18
1	1.010	1.020	1.040	1.060	1.080	1.100	1.120	1.140	1.160	1.180
2	0.507	0.515	0.530	0.545	0.561	0.576	0.592	0.607	0.623	0.639
3	0.340	0.347	0.360	0.374	0.388	0.402	0.416	0.431	0.445	0.460
5	0.206	0.212	0.225	0.237	0.251	0.264	0.277	0.291	0.305	0.320
10	0.106	0.111	0.123	0.136	0.149	0.163	0.177	0.192	0.207	0.222
15	0.072	0.078	0.090	0.103	0.117	0.132	0.147	0.163	0.179	0.196
20	0.055	0.061	0.074	0.087	0.102	0.117	0.134	0.151	0.169	0.187
25	0.045	0.051	0.064	0.078	0.094	0.110	0.128	0.145	0.164	0.183
30	0.039	0.045	0.058	0.073	0.089	0.106	0.124	0.143	0.162	0.181
40	0.030	0.037	0.051	0.067	0.084	0.102	0.121	0.141	0.160	0.180
50	0.026	0.032	0.047	0.063	0.082	0.101	0.120	0.140	0.160	0.180
75	0.019	0.026	0.042	0.061	0.080	0.100	0.120	0.140	0.160	0.180
100	0.016	0.023	0.041	0.060	0.080	0.100	0.120	0.140	0.160	0.180

Key: n = number of years i = interest rate, %

For uniform (equal) future payments each of value F , Equation (3.5) becomes

$$P = \frac{F}{r} \quad \text{or} \quad r = \frac{F}{P} \quad (3.9)$$

If the interest is calculated continuously, rather than periodically, the equivalent of Equation (3.5) is (with the uniform payments of value F)

$$P = F \frac{e^{in} - 1}{i(e^{in})} \quad (3.10)$$

The inverse of the right-hand side of Equation (3.6) is known in economics as the “sinking fund deposit factor,” that is, how much a borrower must periodically deposit with a trustee to eventually pay off a loan.

Now let us look at some examples that illustrate the application of the concepts and relations discussed earlier.

EXAMPLE 3.4 PAYING OFF A LOAN

You borrow \$35,000 from a bank at 10.5% interest to purchase a multicone cyclone rated at 50,000 ft³/min. If you make monthly payments of \$325 (at the end of the month), how many payments will be required to pay off the loan?

Solution The diagram on the time line in Figure E3.4a shows the cash flows. Because the payments are uniform, we can use Equation (3.5), but use \$325 per month rather than \$1.

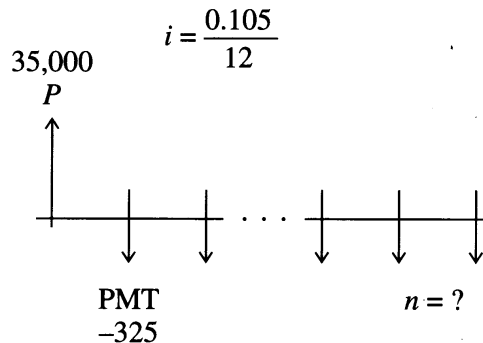


FIGURE E3.4a

$$35,000 - 325 \left[\frac{(1 + i)^n - 1}{i(1 + i)^n} \right] = 0 \quad (a)$$

Equation (a) can be solved for n (months). Use Equation (3.8) to simplify the procedure.

$$r = \frac{i(i + 1)^n}{(1 + i)^n - 1}$$

$$(i + 1)^n = \frac{r}{r - i}$$

$$n = \frac{\ln [r/(r - 1)]}{\ln(1 + i)} \quad (b)$$

In the example the data are

$$i = \frac{0.105}{12} = 0.008750 \quad 1 + i = 1.008750$$

$$r = \frac{325}{35,000} = 0.009286 \quad r/(r - i) = 17.3333$$

$$n = \frac{2.85263}{0.008712} = 327.4 \text{ months}$$

The final payment (No. 328) will be less than \$325.00, namely \$143.11.

For income tax purposes, you can calculate the principal and interest in each payment. For example, at the end of the first month, the interest paid is \$35,000 (0.008750) = \$306.25 and the principal paid is \$325.00 - \$306.25 = \$18.75, so that the principal balance for the next month's interest calculation is \$34,981.25. Iteration of this procedure (best done on a computer) yields the "amortization schedule" for the loan.

You can carry out the calculations using the Microsoft Excel function key (found by clicking on the "insert" button in the toolbar):

1. Click on the function key (f_x) in the spreadsheet tool bar.
2. Choose financial function category (Figure E3.4b).
3. Select NPER.

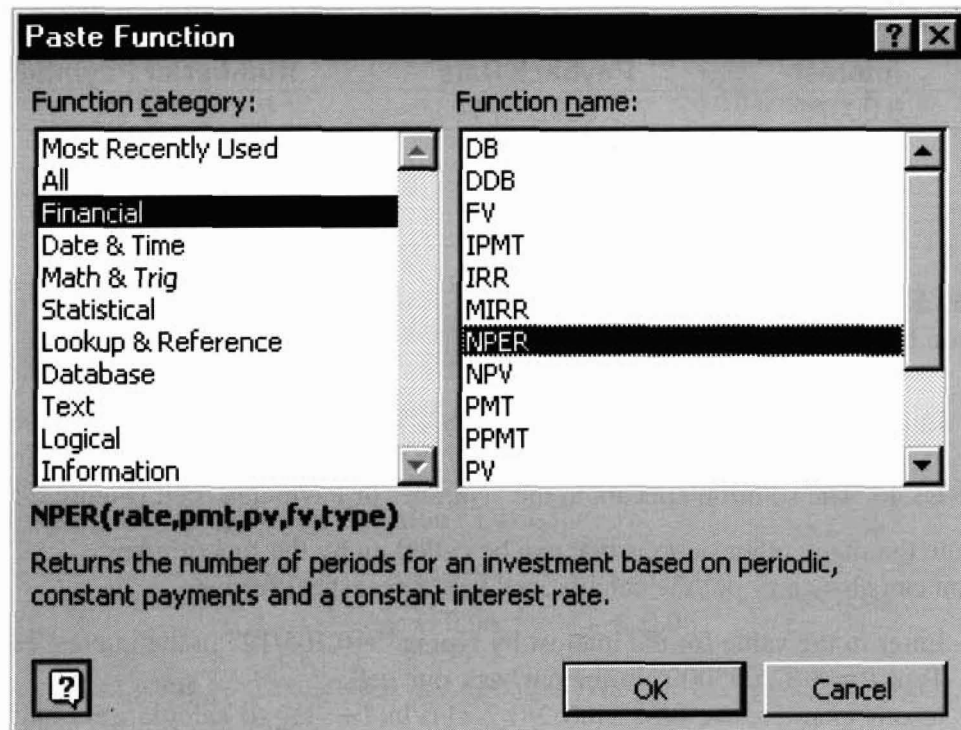


FIGURE E3.4b
Permission by Microsoft.

NPER

Rate = 0.00875

Pmt = -325

Pv = 35000

Fv = number

Type = number

= 327.4392653

Returns the number of periods for an investment based on periodic, constant payments and a constant interest rate.

Pv is the present value, or the lump-sum amount that a series of future payments is worth now.

Formula result = 327.4392653

FIGURE E3.4c

Permission by Microsoft.

Microsoft Excel - loan

File Edit View Insert Format Tools Data Window Help

File Edit View Insert Format Tools Data Window Help

Arial 10 B I U

B13 =

	A	B	C
1	Interest	Payback Rate	Number of Payments
2	0.00875	0.009285714	
3			
4			
5			

FIGURE E3.4d

Permission by Microsoft.

4. Enter correct values for payment ($-\$325$), rate ($0.105/12$), and present value ($\$35,000$) (Figure E3.4c), and click on “OK” to get the screen shown in Figure E3.4d. The solution appears in the “Number of Payments” cell (Figure E3.4e).

Note the many other options that can be called up by the function key. You can also carry out the calculations in a spreadsheet format.

1. Enter in the value for the interest by typing “=0.105/12” in the interest cell.
2. Type “= -325/35000” in the payback rate cell.
3. In our example we type “ln(b2/(b2-a1))/ln(1+a1)” to calculate the number of payments.

	A	B	C
1	Interest	Payback Rate	Number of Payments
2	0.00875	0.009285714	327.4392653
3			
4			
5			

FIGURE E3.4e
Permission by Microsoft.

EXAMPLE 3.5 SELECTION OF THE CHEAPEST ANODES

Ordinary anodes for an electrochemical process last 2 years and then have to be replaced at a cost of \$20,000. An alternative choice is to buy impregnated anodes that last 6 years and cost \$56,000 (see Figure E3.5). If the annual interest rate is 6 percent per year, which alternative would be the cheapest?

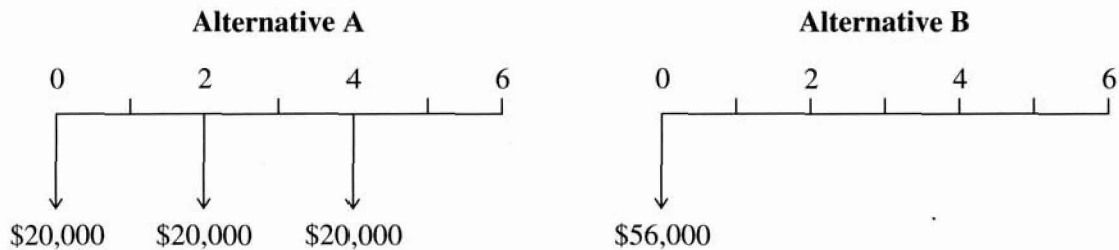


FIGURE E3.5

Solution We want to calculate the present value of each alternative. The present value of alternative A using Equation (3.4) is

$$P = \frac{-\$20,000}{1} + \frac{-\$20,000}{(1 + 0.06)^2} + \frac{-\$20,000}{(1 + 0.06)^4} = -\$53,642$$

The present value of alternative B is $-\$56,000$. Alternative A gives the largest (smallest negative) present value.

3.3 MEASURES OF PROFITABILITY

As mentioned previously, most often in the chemical process industries the objective function for potential projects is some measure of profitability. The projects with highest priorities are the ones with the highest expected profitability; “expected” implies that probabilistic considerations must be taken into account (Palvia and Gordon, 1992), such as calculating the upper and lower bounds of a prediction. In this section, however, we are concerned with a deterministic approach for evaluating profitability, keeping in mind that different definitions of profitability can lead to different priority rankings. Analyses are typically carried out in spreadsheets to generate a variety of possibilities that allow the projects to be ranked as a prelude to decision making.

Among the numerous measures of economic performance that have been proposed, two of the simplest are

1. Payback period (PBP)—how long a project must operate to break even; ignores the time value of money.

$$\text{PBP} = \frac{\text{Cost of investment}}{\text{Cash flow per period}}$$

Example: For an investment of \$20,000 with a return of \$500 per week the PBP is

$$\frac{\$20,000}{\$500} = 40 \text{ weeks}$$

2. Return on investment (ROI)—a simple yield calculation without taking into account the time value of money

$$\text{ROI (in percent)} = \frac{\text{Net income (after taxes) per year}}{\text{Cost of investment}} \times 100$$

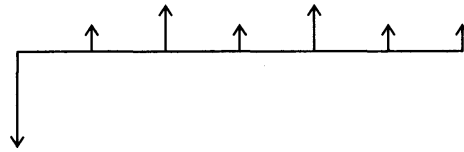
Example: Given the net return of \$6000 (per year) for an initial investment of \$45,000, the ROI is

$$\frac{\$6000}{\$45,000} \times 100 = 13.3\%/\text{year}$$

Two other measures of profitability that take into account the time value of money are

1. Net present value (NPV).
2. Internal rate of return (IRR).

NPV takes into account the size and profitability of a project, but the IRR measures only profitability. If a company has sufficient resources to consider several small projects, given a prespecified amount of investment, a number of high-value IRRs usually provide a higher overall NPV than a single large project.

**FIGURE 3.6**

Cash flows used in calculating net present value (NPV) and internal rate of return (IRR) for a typical capital investment project.

Figure 3.6 designates the cash flows that might occur for a cash investment in a project. NPV is calculated by adding the initial investment (represented as a negative cash flow) to the present value of the anticipated future positive (and negative) cash flows. Equation (3.4) showed how to calculate NPV.

- If the NPV is positive, the investment increases the company's assets: The investment is financially attractive.
- If the NPV is zero, the investment does not change the value of the company's assets: The investment is neutral.
- If the NPV is negative, the investment decreases the company's assets: The investment is not financially attractive.

The higher the NPV among alternative investments with the same capital outlay, the more attractive the investment.

IRR is the rate of return (interest rate, discount rate) at which the future cash flows (positive plus negative) would equal the initial cash outlay (a negative cash flow). The value of the IRR relative to the company standards for internal rate of return indicates the desirability of an investment:

- If the IRR is greater than the designated rate of return, the investment is financially attractive.
- If the IRR is equal to the designated rate of return, the investment is marginal.
- If the IRR is less than the designated rate of return, the investment is financially unattractive.

Table 3.2 compares some of the features of PBP, NPV, and IRR.

Numerous other measures of profitability exist, and most companies (and financial professionals) use more than one. Cut-off levels are placed on the measures of profitability so that proposals that fall below the cut-off level are not deemed worthy of consideration. Those that fall above the cut-off level can be ranked in order of profitability and examined in more detail.

In optimization you are interested in

1. Minimizing the payback period (PBP), or
2. Maximizing the net present value (NPV), or
3. Maximizing the internal rate of return (IRR)

TABLE 3.2
Comparisons of various methods used in economic analyses

Payback period (PBP)	Net present value (NPV)	Internal rate of return (IRR)
Definition		
Number of years for the net after-tax income to recover the net investment without considering time value of money	Present worth of receipts less the present worth of disbursements	IRR equals the interest rate i such that the NPV of receipts less NPV of disbursements equals zero
Advantages		
Measure of fluidity of an investment	Works with all cash flow patterns	Gives rate of return that is a familiar measure and indicates relative merits of a proposed investment
Commonly used and well understood	Easy to compute Gives correct ranking in most project evaluations	Treats variable cash flows Does not require reinvestment rate assumption
Disadvantages		
Does not measure profitability Ignores life of assets	Is not always possible to specify a reinvestment rate for capital recovered	Implicitly assumes that capital recovered can be reinvested at the same rate
Does not properly consider the time value of money and distributed investments or cash flows	Size of NPV (\$) sometimes fails to indicate relative profitability	Requires trial-and-error calculation Can give multiple answers for distributed investments

or optimizing another criterion of profitability. The decision variables are adjusted to reach an extremum. In most of the problems and examples in the subsequent chapters we have not included factors for the time value of money because we want to focus on other details of optimization. Nevertheless, the addition of such factors is quite straightforward.

EXAMPLE 3.6 CALCULATION OF THE OPTIMAL INSULATION THICKNESS

In Example 3.3 we developed an objective function for determining the optimal thickness of insulation. In that example the effect of the time value of money was introduced as an arbitrary constant value of r , the repayment multiplier. In this example, we treat the same problem, but in more detail. We want to determine the optimum insulation thickness for a 20-cm pipe carrying a hot fluid at 260°C. Assume that curvature of the pipe can be ignored and a constant ambient temperature of 27°C exists. The following information applies:

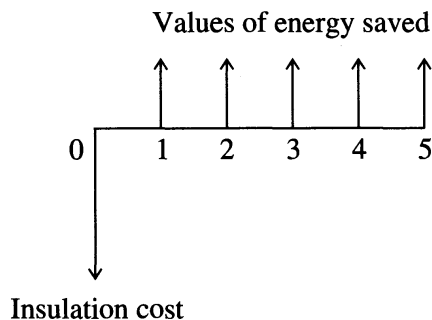


FIGURE E3.6
Cash flows for insulating a pipe.

Y	8000 operating hours/year
H_t	3.80/10 ⁶ kJ fuel cost, 80% thermal efficiency (boiler)
k	0.80 kJ/(h)(m)(°C), insulation
C_1	\$34/cm insulation for 1 m ² of area, cost of insulation
h_c	32.7 kJ/(h)(m ²)(°C), heat transfer coefficient (still air)
	Life of the insulation = 5 years
	Annual discount rate (i) = 14%
L	100 m, length of pipe

The insulation of thickness x can be purchased in increments of 1 cm (i.e., 1, 2, 3 cm, etc.). Equation (b) in Example 3.3 still applies. The value of the energy saved each year over 5 years is

$$Q_0 - Q = \Delta T(\pi DL) \left[h_c - \frac{1}{(x/k) + (1/h_c)} \right] (Y)(H_t) \quad \text{in \$/year}$$

and the cost of the insulation is

$$C_1 x(\pi DL) \quad \text{in \$}$$

at the beginning of the 5-year period. Figure E3.6 is the time line on which the cash flows are placed.

The basis for the calculations will be $L = 100\text{m}$. Because the insulation comes in 1-cm increments, let us calculate the net present value of insulating the pipe as a function of the independent variable x ; vary x for a series of 1-, 2-, 3-cm (etc.) thick increments to get the respective internal rates of return, the payback period, and the return on investment. The latter two calculations are straightforward because of the assumption of five even values for the fuel saved. The net present value and internal rates of return can be compared for various thicknesses of insulation. The cost of the insulation is an initial negative cash flow, and a sum of five positive values represent the value of the heat saved. For example, for 1 cm insulation the net present value is ($r = 0.291$ from Table 3.1)

$$P_1 = -\$2135 + \frac{\$5281}{0.291} = \$16,013$$

A summary of the calculations is

Insulation thickness x (cm)	Insulation cost (\$)	Value of fuel saved (\$/year)	Payback period (years)	Return on investment (% per year)	Net present value (\$)	Internal rate of return (%)
1	2,135	5,281	1.27	79	16,013	247
2	4,270	8,182	1.64	61	23,847	191
3	6,405	10,020	2.01	50	28,028	155
4	8,540	11,288	2.38	42	30,250	130
5	10,675	12,215	2.75	36	31,301	112
6	12,810	12,984	3.10	32	31,809	98
7	14,945	13,480	3.48	29	31,378	86

From Example 3.3, Equation E3.3(d) gives $x \approx 6.4$ cm as the optimal thickness corresponding to the net present value as the criterion for selection. Note that the optimal thickness chosen depends on the criterion you select.

Additional examples of the use of PBP, NPV, and IRR can be found in Appendix B. In Section B.5, we present a more detailed explanation of the various components that constitute the income and expense values that must be used in project evaluation.

REFERENCES

- Brummerstedt, E. F. *Natl Pet News* **36**: R282, R362, R497 (1944).
- Garrett, D. E. *Chemical Engineering Economics*. Van Nostrand-Reinhold, New York (1989).
- Happel, J.; and D. G. Jordan. *Chemical Process Economics*, 2d ed. Marcel Dekker, New York (1975).
- Hurvich, C.; and C. L. Tsai. "A Corrected Akaike Information Criterion for Vector Autoregressive Model Selection." *J Time Series Anal* **14**, 271–279 (1993).
- Kamimura, R. "D-Entropy Minimization." In *Proceedings of the International Conference on Neural Networks*. Houston (1997).
- Palvia, S. C.; and S. R. Gordon. "Tables, Trees, and Formulas in Decision Analysis." *Commun A C M* **35**, 104–113 (1992).
- Rusnak, I.; A. Guez; and I. Bar-Kana. "Multiple Objective Approach to Adaptive Control of Linear Systems." In *Proceedings of the American Control Conference*. San Francisco, pp. 1101–1105 (1993).
- Steuer, R. E. *Multiple Criteria Optimization: Theory, Computation and Application*. Wiley, New York (1998).

SUPPLEMENTARY REFERENCES

- Bhaskar, V.; S. K. Gupta; and A. K. Ray. "Applications of Multiobjective Optimization in Chemical Engineering." *Reviews in Chem Engr* **16** (1): 1–54 (2000).

- Blank, L. T.; and A. J. Tarquin. *Engineering Economy*. McGraw-Hill, New York (1997).
 Bowman, M. S. *Applied Economic Analysis for Technologists, Engineers, and Managers*.
 Prentice-Hall, Englewood Cliffs, NJ (1998).
 Canada, J. R.; W. G. Sullivan; and J. A. White. *Capital Investment Analysis for Engineering
 and Management*. Prentice-Hall, Englewood Cliffs, NJ (1995).

PROBLEMS

- 3.1 If you borrow \$100,000 from a lending agency at 10 percent yearly interest and wish to pay it back in 10 years in equal installments paid annually at the end of the year, what will be the amount of each yearly payment? Compute the principal and interest payments for each year.
- 3.2 Compare the present value of the two depreciation schedules listed below for $i = 0.12$ and $n = 10$ years. Depreciation is an expense and thus has a negative sign before each value. The present value also have a negative sign.

Year	(a)	(b)
1	-1000	-800
2	-1000	-1400
3	-1000	-1200
4	-1000	-1000
5	-1000	-1000
6	-1000	-1000
7	-1000	-900
8	-1000	-900
9	-1000	-900
10	-1000	-900

- 3.3 To provide for the college education of a child, what annual interest rate must you obtain to have a current investment of \$5000 grow to become \$10,000 in 8 years if the interest is compounded annually?
- 3.4 A company is considering a number of capital improvements. Among them is purchasing a small pyrolysis unit that is estimated to earn \$15,000 per year at the end of each year for the next 5 years at which time the sellers agree to purchase the unit back for \$550,000. Ignore tax effects, risk, and so on, and determine the present value of the investment based on an interest rate of 15.00% compounded annually. At the end of year 2 there will be an expense of \$25,000 to replace the unit combustion chamber.
- 3.5 One member of your staff suggests that if your department spends just \$10,000 to improve a process, it will yield cost savings of \$3000, \$5000, and \$4000 over the next 3 years, respectively, for a total of \$12,000. Your company policy is to have an internal rate of return of at least 15% on process improvements. What is the NPV of this proposed improvement?
- 3.6 You want to save for a cruise in the Caribbean. If you place in a savings account at 6% interest \$200 at the beginning of the first year, \$350 at the beginning of the next year, and

\$250 at the beginning of the third year, how much will you have available at the end of the third year?

- 3.7** You open a savings account today (the middle of the month) with a \$775 deposit. The account pays $6\frac{1}{4}\%$ interest (annual value) compounded semimonthly. If you make semimonthly deposits of \$50 beginning next month, how long will it take for your account to reach \$4000?
- 3.8** Looking forward to retirement, you wish to accumulate \$60,000 after 15 years by making deposits in an account that pays $9\frac{3}{4}\%$ interest compounded semiannually. You open the account with a deposit of \$3200 and intend to make semiannual deposits, beginning 6 months later, from your profit-sharing bonus paychecks. Calculate how much these deposits should be.
- 3.9** What is the present value of the tax savings on the annual interest payments if the loan payments consist of five equal monthly installments of principal and interest of \$3600 on a loan of \$120,000. The annual interest rate is 14.0%, and the tax rate is 40%. (Assume the loan starts at the first of July so that only five payments are made during the year on the first of each month starting August 1.)
- 3.10** The following advertisement appeared in the newspaper. Determine whether the statement in the ad is true or false, and show by calculations or explanation why your answer is correct.

A 15-year fixed-rate mortgage with annual payments saves you nearly 60 percent of the total interest costs over the life of the loan compared with a 30-year fixed-rate mortgage.

- 3.11** You borrow \$300,000 for 4 years at an interest rate of 10% per year. You plan to pay in equal annual, end-of-year installments. Fill in the following table.

Year	Balance due at beginning of year, \$	Principal payment, \$	Interest payment, \$	Total payment, \$
1				
2				
3				
4				

- 3.12** Consideration is being given to two plans for supplying water to a plant. Plan A requires a pipeline costing \$160,000 with annual operation and upkeep costs of \$2200, and an estimated life of 30 years with no salvage. Plan B requires a flume costing \$34,000 with a life of 10 years, a salvage value of \$5600, and annual operation and upkeep of \$4500 plus a ditch costing \$58,000, with a life of 30 years and annual costs for upkeep of \$2500. Using an interest rate of 12 percent, compare the net present values of the two alternatives.
- 3.13** Cost estimators have provided reliable cost data as shown in the following table for the chlorinators in the methyl chloride plant addition. Analysis of the data and recommendations of the two alternatives are needed. Use present worth for $i = 0.10$ and $i = 0.20$.

	Chlorinators	
	Glass-lined	Cast iron
Installed cost	\$24,000	\$7200
Estimated useful life	10 years	4 years
Salvage value	\$4000	\$800
Miscellaneous annual costs as percent of original cost	10	20
Maintenance costs		
<i>Glass-lined.</i> \$230 at the end of the second year, \$560 at the end of the fifth year, and \$900 at the end of each year thereafter.		
<i>Cast iron.</i> \$730 each year.		

The product from the glass-lined chlorinator is essentially iron-free and is estimated to yield a product quality premium of \$1700 per year. Compare the two alternatives for a 10-year period. Assume the salvage value of \$800 is valid at 10 years.

- 3.14 Three projects (A, B, C) all earn a total of \$125,000 over a period of 5 years (after-tax earnings, nondiscounted). For the cash-flow patterns shown in the table, predict by inspection which project will have the largest rate of return. Why?

Year	Cash flow, \$10 ³		
	A	B	C
1	45	25	10
2	35	25	30
3	25	25	45
4	15	25	30
5	5	25	10

- 3.15 Suppose that an investment of \$100,000 will earn after-tax profits of \$10,000 per year over 20 years. Due to uncertainties in forecasting, however, the projected after-tax profits may be in error by ± 20 percent. Discuss how you would determine the sensitivity of the rate of return to an error of this type. Would you expect the rate of return to increase by 20 percent of its computed value for a 20-percent increase in annual after-tax profits (i.e., to \$12,000)?
- 3.16 The installed capital cost of a pump is \$200/hp and the operating costs are 4¢/kWh. For 8000 h/year of operation, an efficiency of 70 percent, and a cost of capital $i = 0.10$, for $n = 5$ years, determine the relative importance of the capital versus operating costs.
- 3.17 The longer it takes to build a facility, the lower its rate of return. Formulate the ratio of total investment I divided by annual cash flow C (profit after taxes plus depreciation) in terms of 1-, 2-, and 3-year construction periods if $i =$ interest rate, and $n =$ life of facility (no salvage value).
- 3.18 A chemical valued at \$0.94/lb is currently being dried in a fluid-bed dryer that allows 0.1 percent of the 4-million lb/year throughput to be carried out in the exhaust. An engineer is considering installing a \$10,000 cyclone that would recover the fines; extra

pressure drop is no concern. What is the expected payback period for this investment? Maintenance costs are estimated to be \$300/year. The inflation rate is 8 percent, and the interest rate 15 percent.

- 3.19** To reduce heat losses, the exterior flat wall of a furnace is to be insulated. The data presented to you are

Temperature inside the furnace at the wall	500°F (constant)
Air temperature outside wall	Assume constant at 70°F
Heat transfer coefficients	
Outside air film (h)	4 Btu/(h)(ft ²)(°F)
Conductivity of insulation (k)	0.03 Btu/(hr)(ft)(°F)
Cost of insulation	\$0.75/(ft ²) (per inch of thickness)
Values of energy saved	\$0.60/10 ⁶ Btu
Hours of operation	8700/year
Interest rate	30% per year for capital costs

Note that the overall heat transfer coefficient U is related to h and k by

$$\frac{1}{U} = \frac{1}{h} + \frac{t}{(12)(k)}$$

where t is the thickness in inches of the insulation, and the heat transfer through the wall is $Q = UA(T_{\text{furnace}} - T_{\text{wall}})$, where T is in °F. Ignore any effect of the uninsulated part of the wall.

What is the minimum cost for the optimal thickness of the insulation? List specifically the objective function, all the constraints, and the optimal value of t . Show each step of the solution. Ignore the time value of money for this problem.

- 3.20** We want to optimize the heat transfer area of a steam generator. A hot oil stream from a reactor needs to be cooled, providing a source of heat for steam production. As shown in Figure P3.20, the hot oil enters the generator at 400°F and leaves at an unspecified temperature T_2 ; the hot oil transfers heat to a saturated liquid water stream at 250°F, yielding steam (30 psi, 250°F). The other operating conditions of the exchanger are

$$U = 100 \text{ Btu}/(\text{h})(\text{ft}^2)(\text{°F}) \quad \text{overall heat transfer coefficient}$$

$$w_{\text{oil}}C_{p_{\text{oil}}} = 7.5 \times 10^4 \text{ Btu}/(\text{°F})(\text{h})$$

We ignore the cost of the energy of pumping and the cost of water and only consider the investment cost of the heat transfer area. The heat exchanger cost is \$25/ft² of heat

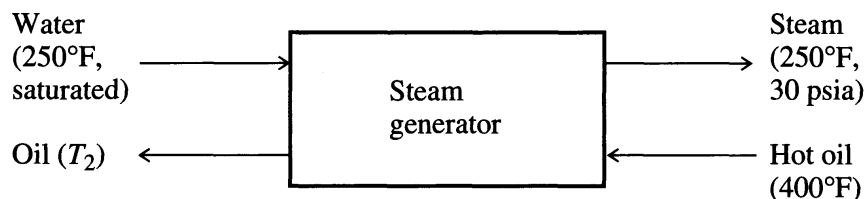


FIGURE P3.20
Steam generator flow diagram.

transfer surface. You can expect a credit of $\$2/10^6$ Btu for the steam produced. Assume the exchanger will be in service 8000 h/year. Find the outlet temperature T_2 and heat exchanger area A that maximize the profitability, as measured by (a) return on investment (ROI) and (b) net present value.

3.21 In *Chemical Engineering* (Jan. 1994, p. 103) the following explanation of internal rate of return appeared:

Internal return rate. *The internal return rate (IRR), also known as the discounted cash flow return rate, is the iteratively calculated discounting rate that would make the sum of the annual cash flows, discounted to the present, equal to zero. As shown in Figure 2, the IRR for Project Chem-A is 38.3%/yr. Note that this single fixed point represents the zero-profitability situation. It does not vary with the cost of capital (discount rate), although the profitability should increase as the cost of capital decreases. There is no way that the IRR can be related to the profitability of a project at meaningful discount rates because of the nonlinear nature of the discounting step.*

What is correct and incorrect about this explanation? Be brief!

3.22 Refer to Problem 3.5. The same staff member asks if the internal rate of return on the proposed project is close to 15%. Calculate the IRR.

3.23 The cost of a piece of equipment is \$30,000. It is expected to yield a cash return per month of \$1000. What is the payback period?

3.24 After retrofitting an extruder, the net additional income after taxes is expected to be \$5000 per year. The remodeling cost was \$50,000. What is the return on investment in percent?

3.25 Your minimum acceptable rate of return (MARR) is 18%, the project life is 10 years, and no alternatives have a salvage value. The following mutually exclusive alternatives have been proposed. Rank them, and recommend the best alternative.

	A	B	C	D	E
Capital investment, \$	38,000	50,000	55,000	60,000	70,000
Net annual earnings, \$	11,000	14,100	16,300	16,800	19,200
IRR, %	26.1	25.2	26.9	25.0	24.3

3.26 You have four choices of equipment (as shown in the following table) to solve a pollution control problem. The choices are mutually exclusive and you must pick one. Assuming a useful life of 10 years for each design, no market value, and a pretax minimum acceptable rate of return (MARR) of 15% per year, rank them and recommend a choice.

Alternative	D_1	D_2	D_3	D_4
Capital investment, \$1000	600	760	1,240	1,600
Annual expenses, \$1000	780	728	630	574
P (present value), \$1000	-\$4,515	-\$4,414	-\$4,402	-\$4,481

3.27 A company invests \$1,000,000 in a new control system for a plant. The estimated annual reduction in cost is calculated to be \$162,000 in each of the next 10 years. What is the

- (a) Return on investment (ROI)
- (b) Internal rate of return (IRR)

Ignore income tax effects and depreciation to simplify the calculations.

3.28 The following table gives a comparison of costs for two types of heaters to supply heat to an oil stream in a process plant at a rate of 73,500,000 Btu/h:

	Oil convection	Rotary air preheater
Heat input in 10^6 Btu/h	114.0	96.5
Thermal efficiency, %	64.5	76.1
Total fuel cost (at \$1.33/per 10^6 Btu) for 1 year	\$1,261,000	\$1,068,000
Power at \$0.06/kWh for 1 year		48,185
Capital cost (installed), \$	\$1,888,000	\$2,420,000

Assume that the plant in which this equipment is installed will operate 10 years, that a tax rate of 34%/year is applicable, and that a charge of 10% of the capital cost per year for depreciation will be employed over the entire 10-year period, that fixed charges including maintenance incurred by installation of this equipment will amount to 10%/year of the investment, and that a minimum acceptable return rate on invested capital after taxes and depreciation is 15%. Determine which of the two alternative installations should be selected, if any.

3.29 You are proposing to buy a new, improved reboiler for a distillation column that will save energy. You estimate that the initial investment will be \$140,000, annual savings will be \$25,000 per year, the useful life will be 12 years, and the salvage value at the end of that time will be \$40,000. You are ignoring taxes and inflation, and your pretax constant dollar minimum acceptable rate of return (MARR) is 10% per year. Your boss wants to see a sensitivity diagram showing the present worth as a function of $\pm 50\%$ changes in annual savings and the useful life.

- (a) What is the present value P of your base case?
- (b) You calculate the P of -50% annual savings to be $-\$42,084$ and the P for $+50\%$ annual savings to be $\$128,257$. The P at -50% life is $-\$8,539$. What is the P at $+50\%$ life?
- (c) Sketch the P sensitivity diagram for these two variables [P vs the change in the base (in %)]. To which of the two variables is the decision most sensitive?

PART II

OPTIMIZATION THEORY AND METHODS

PART II DESCRIBES modern techniques of optimization and translates these concepts into computational methods and algorithms. Because the literature on optimization techniques is vast, we focus on methods that have proved effective for a wide range of problems. Optimization methods have matured sufficiently during the past 20 years so that fast and reliable methods are available to solve each important class of problem.

Seven chapters make up Part II of this book, covering the following areas:

1. Mathematical concepts (Chapter 4)
2. One-dimensional search (Chapter 5)
3. Unconstrained multivariable optimization (Chapter 6)
4. Linear programming (Chapter 7)
5. Nonlinear programming (Chapter 8)
6. Optimization involving discrete variables (Chapter 9)
7. Global optimization (Chapter 10)

The topics are grouped so that unconstrained methods are presented first, followed by constrained methods. The last two chapters in Part II deal with discontinuous (integer) variables, a common category of problem in chemical engineering, but one quite difficult to solve without great effort.

As optimization methods as well as computer hardware and software have improved over the past two decades, the degree of difficulty of the problems that can be solved has expanded significantly. Continued improvements in optimization algorithms and computer technology should enable optimization of large-scale nonlinear problems involving thousands of variables, both continuous and integer, some of which may be stochastic in nature.

BASIC CONCEPTS OF OPTIMIZATION

4.1 Continuity of Functions 114
4.2 NLP Problem Statement 118
4.3 Convexity and Its Applications 121
**4.4 Interpretation of the Objective Function in Terms of Its Quadratic
Approximation 131**
**4.5 Necessary and Sufficient Conditions for an Extremum of an Unconstrained
Function 135**
References 142
Supplementary References 142
Problems 142

TO UNDERSTAND THE strategy of optimization procedures, certain basic concepts must be described. In this chapter we examine the properties of objective functions and constraints to establish a basis for analyzing optimization problems. We identify those features that are desirable (and also undesirable) in the formulation of an optimization problem. Both qualitative and quantitative characteristics of functions are described. In addition, we present the necessary and sufficient conditions to guarantee that a supposed extremum is indeed a minimum or a maximum.

4.1 CONTINUITY OF FUNCTIONS

In carrying out analytical or numerical optimization you will find it preferable and more convenient to work with continuous functions of one or more variables than with functions containing discontinuities. Functions having continuous derivatives are also preferred. Case *A* in Figure 4.1 shows a discontinuous function. Is case *B* also discontinuous?

We define the property of continuity as follows. A function of a single variable x is continuous at a point x_0 if

$$f(x_0) \text{ exists}$$

$$\lim_{x \rightarrow x_0} f(x) \text{ exists}$$

$$\lim_{x \rightarrow x_0} f(x) = f(x_0)$$

If $f(x)$ is continuous at every point in region R , then $f(x)$ is said to be continuous throughout R . For case *B* in Figure 4.1, the function of x has a “kink” in it, but $f(x)$ does satisfy the property of continuity. However, $f'(x) \equiv df(x)/dx$ does not. Therefore, the function in case *B* is continuous but not continuously differentiable.

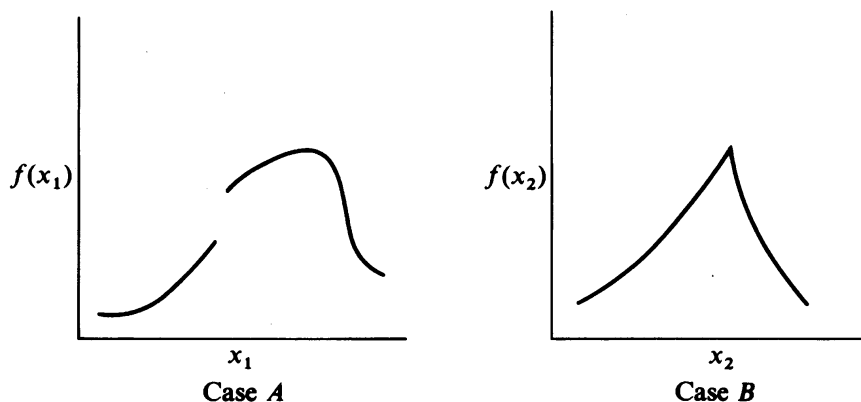


FIGURE 4.1

Functions with discontinuities in the function or derivatives.

EXAMPLE 4.1 ANALYSIS OF FUNCTIONS FOR CONTINUITY

Are the following functions continuous? (a) $f(x) = 1/x$; (b) $f(x) = \ln x$. In each case specify the range of x for which $f(x)$ and $f'(x)$ are continuous.

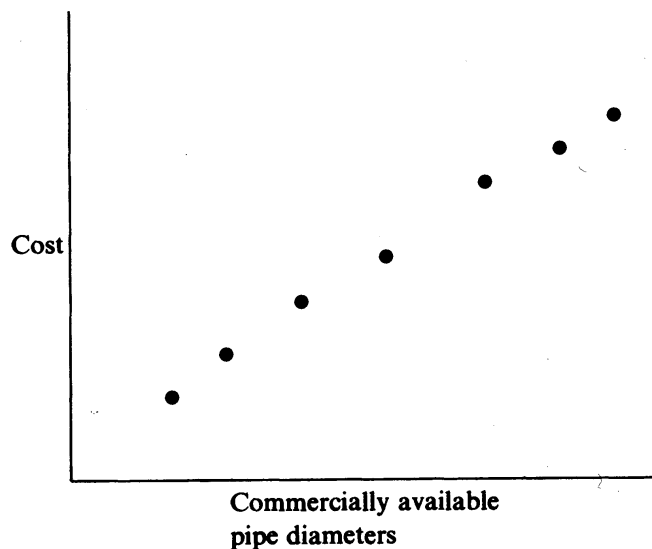
Solution

(a) $f(x) = 1/x$ is continuous except at $x = 0$; $f(0)$ is not defined. $f'(x) = -1/x^2$ is continuous except at $x = 0$.

(b) $f(x) = \ln x$ is continuous for $x > 0$. For $x \leq 0$, $\ln(x)$ is not defined. As to $f'(x) = 1/x$, see (a).

A discontinuity in a function may or may not cause difficulty in optimization. In case *A* in Figure 4.1, the maximum occurs reasonably far from the discontinuity which may or may not be encountered in the search for the optimum. In case *B*, if a method of optimization that does not use derivatives is employed, then the “kink” in $f(x)$ is probably unimportant, but methods employing derivatives might fail, because the derivative becomes undefined at the discontinuity and has different signs on each side of it. Hence a search technique approaches the optimum, but then oscillates about it rather than converges to it.

Objective functions that allow only discrete values of the independent variable(s) occur frequently in process design because the process variables assume only specific values rather than *continuous* ones. Examples are the cost per unit diameter of pipe, the cost per unit area for heat exchanger surface, or the insulation cost considered in Example 1.1. For a pipe, we might represent the installed cost as a function of the pipe diameter as shown in Figure 4.2 [see also Noltie (1978)]. For

**FIGURE 4.2**

Installed pipe cost as a function of diameter.

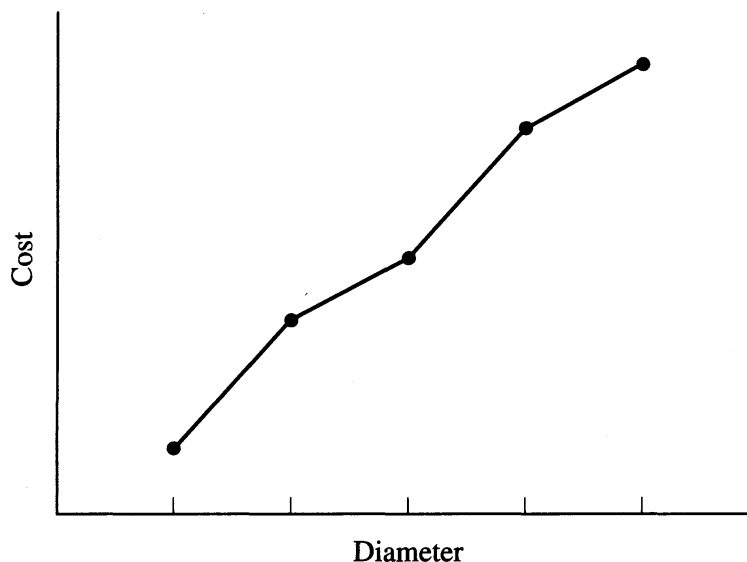


FIGURE 4.3
Piecewise linear approximation to cost function.

most purposes such a cost function can be approximated as a continuous function because of the relatively small differences in available pipe diameters. You can then disregard the discrete nature of the function and optimize the cost as if the diameter were a continuous variable. For example, extend the function of Figure 4.2 to a continuous range of diameters by interpolation. If linear interpolation is used, then the extended function usually has discontinuous derivatives at each of the original diameters, as shown in Figure 4.3. As mentioned earlier, this step can cause problems for derivative-based optimizers. A remedy is to interpolate with quadratic or cubic functions chosen so that their first derivatives are continuous at the break points. Such functions are called splines (Bartela et al., 1987). Once the optimum value of the diameter is obtained for the continuous function, the discretely valued diameter nearest to the optimum that is commercially available can be selected. A suboptimal value for installed cost results, but such a solution should be adequate for engineering purposes because of the narrow intervals between discrete values of the diameter.

EXAMPLE 4.2 OPTIMIZATION INVOLVING AN INTEGER-VALUED VARIABLE

Consider a catalytic regeneration cycle in which there is a simple trade-off between costs incurred during regeneration and the increased revenues due to the regenerated catalyst. Let x_1 be the number of days during which the catalyst is used in the reactor and x_2 be the number of days for regeneration. The reactor start-up crew is only available in the morning shift, so $x_1 + x_2$ must be an integer.

We assume that the reactor feed flow rate q (kg/day) is constant as is the cost of the feed C_1 (\$/kg), the value of the product C_2 (\$/kg), and the regeneration cost C_3

(\$/regeneration cycle). We further assume that the catalyst deteriorates gradually according to the linear relation

$$d = 1.0 - kx_1$$

where 1.0 represents the weight fraction conversion of feed at the start of the operating cycle, and k is the deterioration factor in units of weight fraction per day. Define an objective function and find the optimal value of x_1 .

Solution. For one complete cycle of operation and regeneration, the objective function for the total profit per day comprises

$$\begin{aligned} \frac{\text{Profit}}{\text{Day}} &= \text{Product value} - \text{Feed cost} \\ &\quad - (\text{Regeneration cost per cycle}) \cdot (\text{Cycles per day}) \end{aligned}$$

or in the defined notation

$$f(\mathbf{x}) = \frac{qC_2x_1d_{\text{avg}} - qC_1x_1 - C_3}{x_1 + x_2} \quad (a)$$

where $d_{\text{avg}} = 1.0 - (kx_1/2)$.

The maximum daily profit for an entire cycle is obtained by maximizing Equation (a) with respect to x_1 . As a first trial, we allow x_1 to be a continuous variable. When the first derivative of Equation (a) is set equal to zero and the resulting equation solved for x_1 , the optimum is

$$x_1^{\text{opt}} = -x_2 + \left[x_2^2 + \left(\frac{2}{k} \right) \left(x_2 - \frac{C_1x_2}{C_2} + \frac{C_3}{qC_2} \right) \right]^{1/2}$$

Suppose $x_2 = 2$, $k_1 = 0.02$, $q = 1000$, $C_2 = 1.0$, $C_1 = 0.4$, and $C_3 = 1000$. Then $x_1^{\text{opt}} = 12.97$ (rounded to 13 days if x_1 is an integer).

Clearly, treating x_1 as a continuous variable may be improper if x_1 is 1, 2, 3, and so on, but is probably satisfactory if x_1 is 15, 16, 17, and so on. You might specify x_1 in terms of shifts of 4–8 h instead of days to obtain finer subdivisions of time.

In real life, other problems involving discrete variables may not be so nicely posed. For example, if cost is a function of the number of discrete pieces of equipment, such as compressors, the optimization procedure cannot ignore the integer character of the cost function because usually only a small number of pieces of equipment are involved. You cannot install 1.54 compressors, and rounding off to 1 or 2 compressors may be quite unsatisfactory. This subject will be discussed in more detail in Chapter 9.

4.2 NLP PROBLEM STATEMENT

A general form for a nonlinear program (NLP) is

$$\begin{aligned}
 \text{Minimize:} & \quad f(\mathbf{x}) \\
 \text{Subject to:} & \quad a_i \leq g_i(\mathbf{x}) \leq b_i \quad i = 1, \dots, m \\
 \text{and} & \quad l_j \leq x_j \leq u_j \quad j = 1, \dots, n
 \end{aligned} \tag{4.1}$$

In this problem statement, \mathbf{x} is a vector of n decision variables (x_1, \dots, x_n) , f is the objective function, and the g_i are constraint functions. The a_i and b_i are specified lower and upper bounds on the constraint functions with $a_i \leq b_i$, and l_j, u_j are lower and upper bounds on the variables with $l_j \leq u_j$. If $a_i = b_i$, the i th constraint is an equality constraint. If the upper and lower limits on g_i correspond to $a_i = -\infty$ and $b_i = +\infty$, the constraint is unbounded. Similar comments apply to the variable bounds, with $l_j = u_j$ corresponding to a variable x_j whose value is fixed, and $l_j = -\infty$ and $u_j = +\infty$ specifying a free variable.

Problem 4.1 is nonlinear if one or more of the functions f, g_1, \dots, g_m are nonlinear. It is *unconstrained* if there are no constraint functions g_i and no bounds on the x_i , and it is *bound-constrained* if only the x_i are bounded. In *linearly constrained* problems all constraint functions g_i are linear, and the objective f is nonlinear. There are special NLP algorithms and software for unconstrained and bound-constrained problems, and we describe these in Chapters 6 and 8. Methods and software for solving constrained NLPs use many ideas from the unconstrained case. Most modern software can handle nonlinear constraints, and is especially efficient on linearly constrained problems. A linearly constrained problem with a quadratic objective is called a quadratic program (QP). Special methods exist for solving QPs, and these are often faster than general purpose optimization procedures.

A vector \mathbf{x} is *feasible* if it satisfies all the constraints. The set of all feasible points is called the *feasible region* F . If F is empty, the problem is *infeasible*, and if feasible points exist at which the objective f is arbitrarily large in a max problem or arbitrarily small in a min problem, the problem is *unbounded*. A point (vector) \mathbf{x}^* is termed a local *extremum* (*minimum*) if

$$f(\mathbf{x}^*) \leq f(\mathbf{x}) \tag{4.2}$$

for all \mathbf{x} in a small neighborhood (region) N in F around \mathbf{x}^* with \mathbf{x} distinct from \mathbf{x}^* . Despite the fact that \mathbf{x}^* is a local extremum, other extrema may exist outside the neighborhood N meaning that the NLP problem may have more than one local minimum if the entire space of \mathbf{x} is examined. Another important concept relates to the idea of a *global extremum*, the unique solution of the NLP problem. A *global minimum* occurs if Equation (4.2) holds for all $\mathbf{x} \in F$. Analogous concepts exist for *local maxima* and the *global maximum*. Most (but not all) algorithms for solving NLP problems locate a local extremum from a given starting point.

NLP geometry

A typical feasible region for a problem with two variables and the constraints

$$x_j \geq 0, \quad g_i(\mathbf{x}) \leq 0, \quad i = 1, 2, \quad j = 1, 2$$

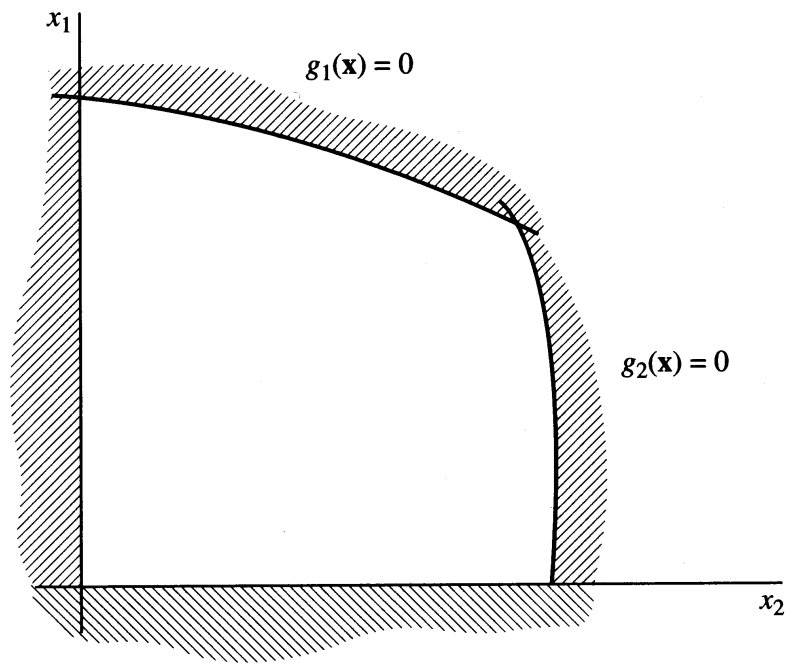


FIGURE 4.4
Feasible region (region not shaded and its boundaries).

is shown as the unshaded region in Figure 4.4. Its boundaries are the straight and curved lines $x_j = 0$ and $g_i(\mathbf{x}) = 0$ for $i = 1, 2, j = 1, 2$.

As another example, consider the problem

$$\text{Minimize } f = (x_1 - 3)^2 + (x_2 - 4)^2$$

subject to the linear constraints

$$x_1 \geq 0$$

$$x_2 \geq 0$$

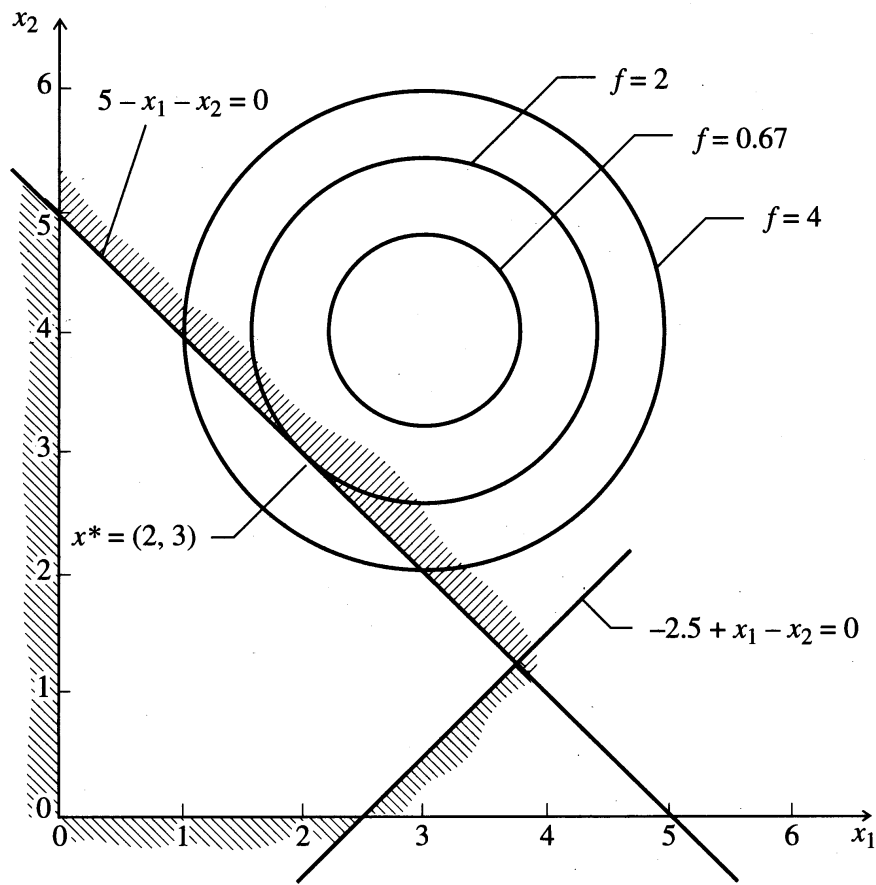
$$5 - x_1 - x_2 \geq 0$$

$$-2.5 + x_1 - x_2 \leq 0$$

This problem is shown in Figure 4.5. The feasible region is defined by linear constraints with a finite number of corner points. The objective function, being non-linear, has contours (the concentric circles, *level sets*) of constant value that are not parallel lines, as would occur if it were linear. The minimum value of f corresponds to the contour of lowest value having at least one point in common with the feasible region, that is, at $x_1^* = 2, x_2^* = 3$. This is not an extreme point of the feasible set, although it is a boundary point. For linear programs the minimum is always at an extreme point, as shown in Chapter 7.

Furthermore, if the objective function of the previous problem is changed to

$$f_1 = (x_1 - 2)^2 + (x_2 - 2)^2$$

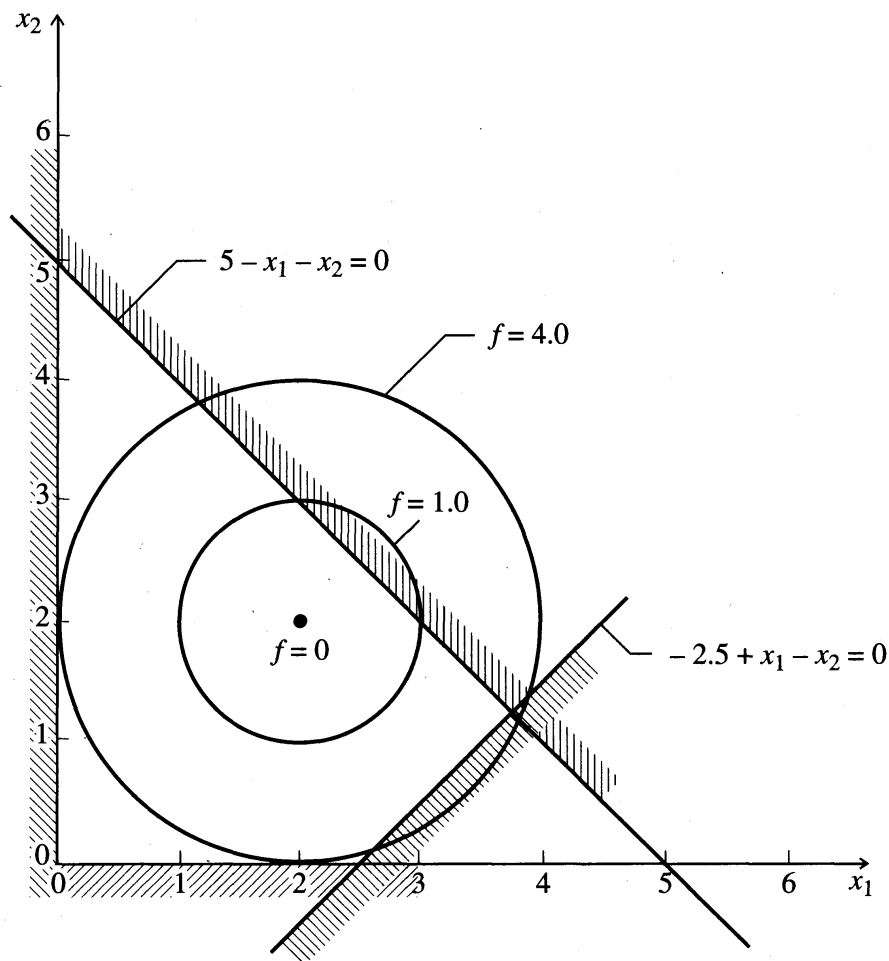
**FIGURE 4.5**

The minimum occurs on the boundary of the constraint set.

as depicted in Figure 4.6, the minimum is now at $x_1 = 2$, $x_2 = 2$, which is not a boundary point of the feasible region, but is the unconstrained minimum of the nonlinear function and satisfies all the constraints.

Neither of the problems illustrated in Figures 4.5 and 4.6 had more than one optimum. It is easy, however, to construct nonlinear programs in which local optima occur. For example, if the objective function f_1 had two minima and at least one was interior to the feasible region, then the constrained problem would have two local minima. Contours of such a function are shown in Figure 4.7. Note that the minimum at the boundary point $x_1 = 3$, $x_2 = 2$ is the global minimum at $f = 3$; the feasible local minimum in the interior of the constraints is at $f = 4$.

Although the examples thus far have involved linear constraints, the chief non-linearity of an optimization problem often appears in the constraints. The feasible region then has curved boundaries. A problem with nonlinear constraints may have local optima, even if the objective function has only one unconstrained optimum. Consider a problem with a quadratic objective function and the feasible region shown in Figure 4.8. The problem has local optima at the two points a and b because no point of the feasible region in the immediate vicinity of either point yields a smaller value of f .

**FIGURE 4.6**

The minimum occurs in the interior of the constraint set.

In summary, the optimum of a nonlinear programming problem is, in general, not at an extreme point of the feasible region and may not even be on the boundary. Also, the problem may have local optima distinct from the global optimum. These properties are direct consequences of nonlinearity. A class of nonlinear problems can be defined, however, that are guaranteed to be free of distinct local optima. They are called convex programming problems and are considered in the following section.

4.3 CONVEXITY AND ITS APPLICATIONS

The concept of *convexity* is useful both in the theory and applications of optimization. We first define a *convex set*, then a *convex function*, and lastly look at the role played by convexity in optimization.

Convex set

A set of points (or a *region*) is defined as a *convex set* in n -dimensional space if, for all pairs of points \mathbf{x}_1 and \mathbf{x}_2 in the set, the straight-line segment joining them is also entirely in the set. Figure 4.9 illustrates the concept in two dimensions.

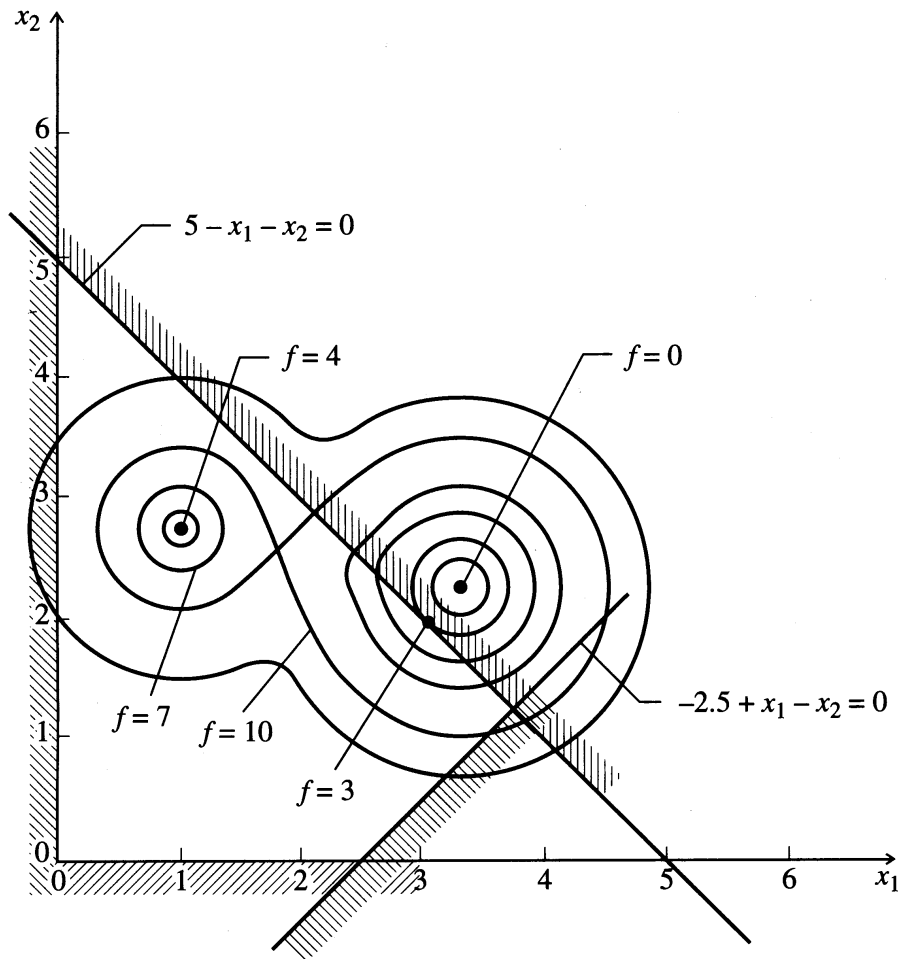


FIGURE 4.7
Local optima due to objective function.

A mathematical statement of a convex set is

For every pair of points \mathbf{x}_1 and \mathbf{x}_2 in a convex set, the point \mathbf{x} given by a linear combination of the two points

$$\mathbf{x} = \gamma\mathbf{x}_1 + (1 - \gamma)\mathbf{x}_2 \quad 0 \leq \gamma \leq 1$$

is also in the set. The convex region may be *closed (bounded)* by a set of functions, such as the sets A and B in Figure 4.9 or may be *open (unbounded)* as in Figures 4.10 and 4.12. Also, the intersection of any number of convex set is a convex set.

Convex function

Next, let us examine the matter of a *convex function*. The concept of a convex function is illustrated in Figure 4.10 for a function of one variable. Also shown is a *concave function*, the negative of a convex function. (If $f(\mathbf{x})$ is convex, $-f(\mathbf{x})$ is concave.) A function $f(\mathbf{x})$ defined on a convex set F is said to be a *convex function* if the following relation holds

$$f[\gamma\mathbf{x}_1 + (1 - \gamma)\mathbf{x}_2] \leq \gamma f(\mathbf{x}_1) + (1 - \gamma)f(\mathbf{x}_2)$$

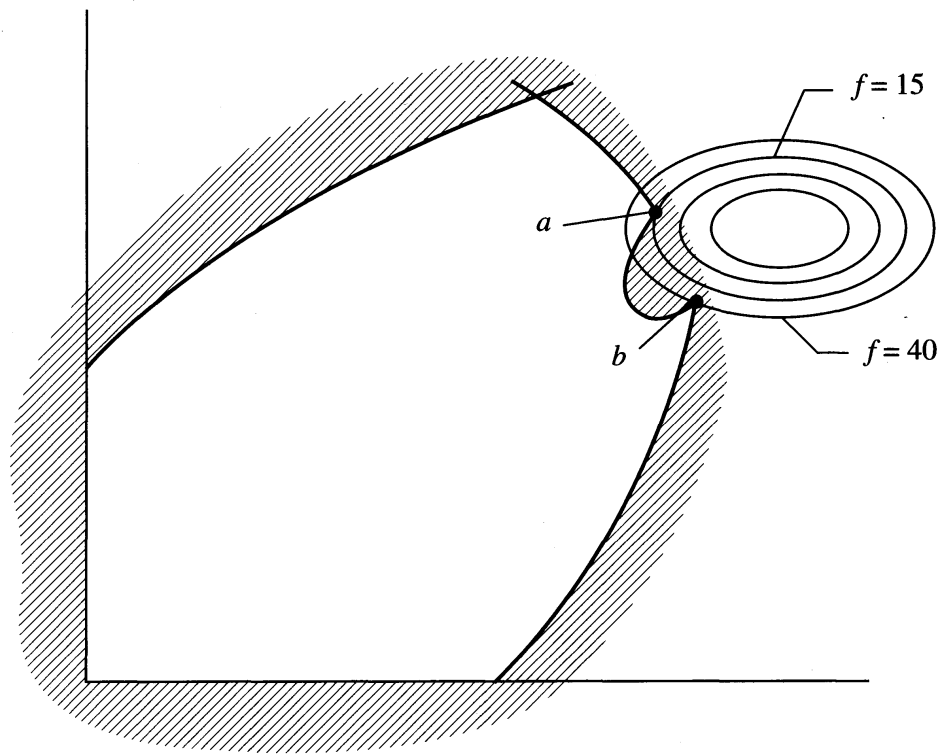


FIGURE 4.8
Local optima due to feasible region.

where γ is a scalar with the range $0 \leq \gamma \leq 1$. If only the inequality sign holds, the function is said to be not only convex but *strictly convex*. [If $f(\mathbf{x})$ is strictly convex, $-f(\mathbf{x})$ is *strictly concave*.] Figure 4.10 illustrates both a strictly convex and a strictly concave function. A convex function cannot have any value larger than the values of the function obtained by linear interpolation between \mathbf{x}_1 and \mathbf{x}_2 (the cord between \mathbf{x}_1 and \mathbf{x}_2 shown in the top figure in Figure 4.10). Linear functions are both convex and concave, but not strictly convex or concave, respectively. An important result of convexity is

If $f(\mathbf{x})$ is convex, then the set

$$R = \{\mathbf{x} | f(\mathbf{x}) \leq k\}$$

is convex for all scalars k .

The result is illustrated in Figure 4.11 in which a convex quadratic function is cut by the plane $f(\mathbf{x}) = k$. The convex set R projected on to the \mathbf{x}_1 - \mathbf{x}_2 plane comprises the boundary ellipse plus its interior.

The convex programming problem

An important result in mathematical programming evolves from the concept of convexity. For the nonlinear programming problem called the *convex programming problem*

$$\begin{aligned} \text{Minimize:} & \quad f(\mathbf{x}) \\ \text{Subject to:} & \quad g_i(\mathbf{x}) \leq 0 \quad i = 1, \dots, m \end{aligned} \quad (4.3)$$

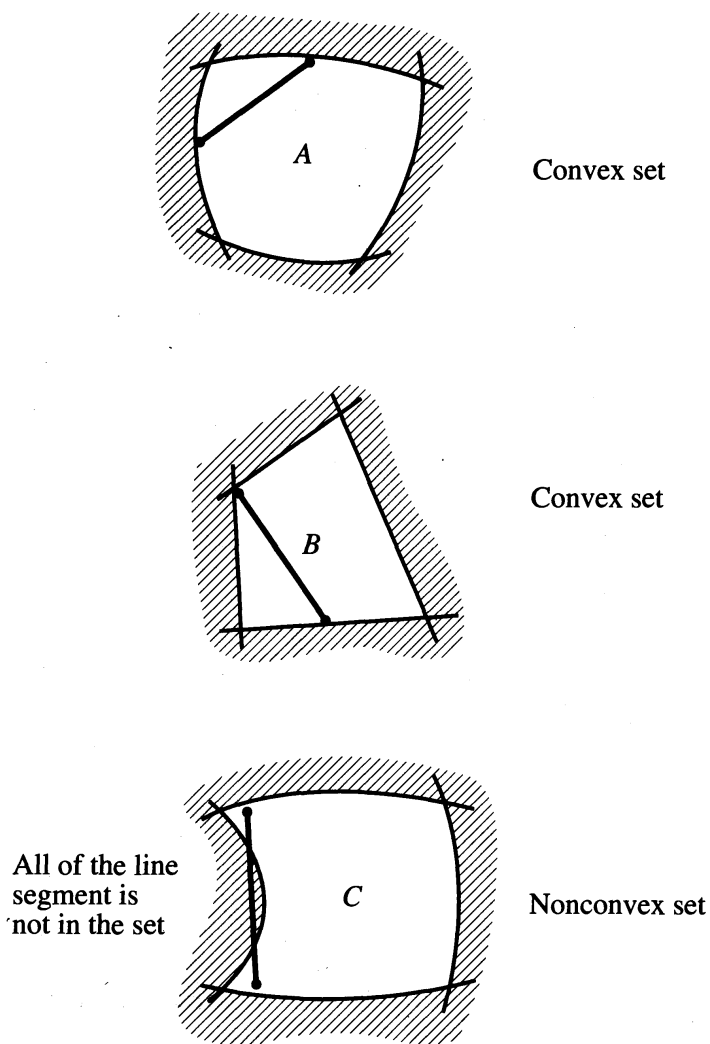


FIGURE 4.9
Convex and nonconvex sets.

in which (a) $f(\mathbf{x})$ is a convex function, and (b) each inequality constraint is a convex function (so that the constraints form a convex set), the following property can be shown to be true

The *local minimum* of $f(\mathbf{x})$ is also the *global minimum*.

Analogously, a local maximum is the global maximum of $f(\mathbf{x})$ if the objective function is concave and the constraints form a convex set.

Role of convexity

If the constraint set $g(\mathbf{x})$ is nonlinear, the set

$$R = \{\mathbf{x} | g(\mathbf{x}) = \mathbf{0}\}$$

is generally not convex. This is evident geometrically because most nonlinear functions have graphs that are curved surfaces. Hence the set R is usually a curved surface also, and the line segment joining any two points on this surface generally does not lie on the surface.

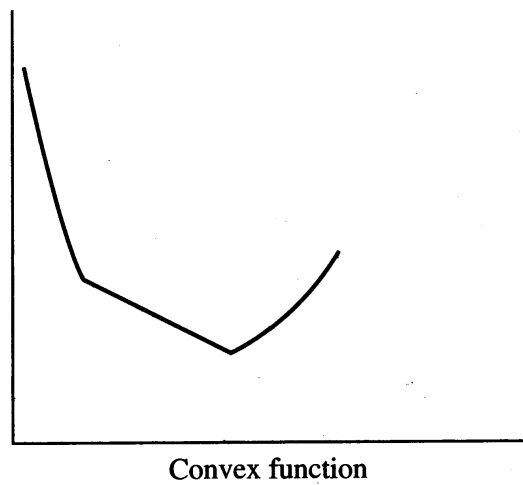
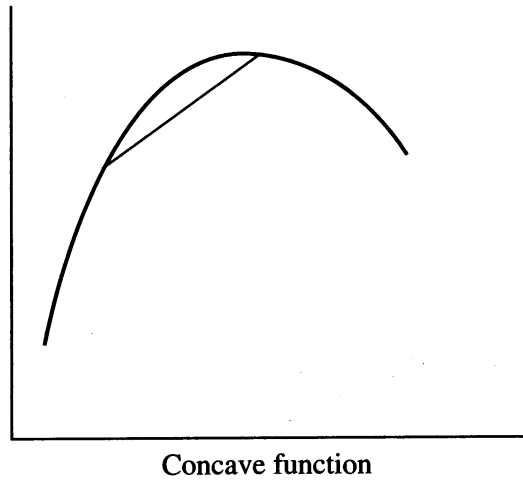
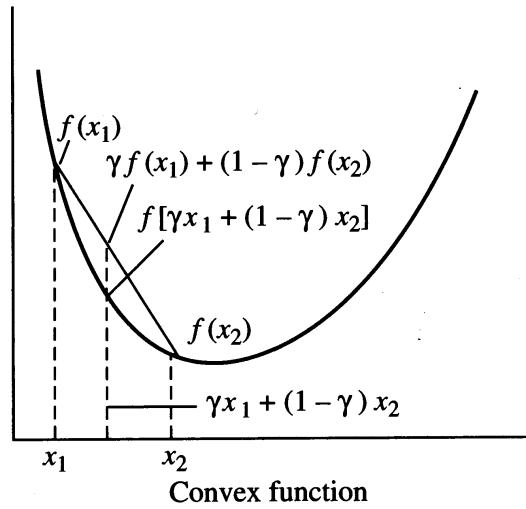


FIGURE 4.10
Convex and concave functions of one variable.

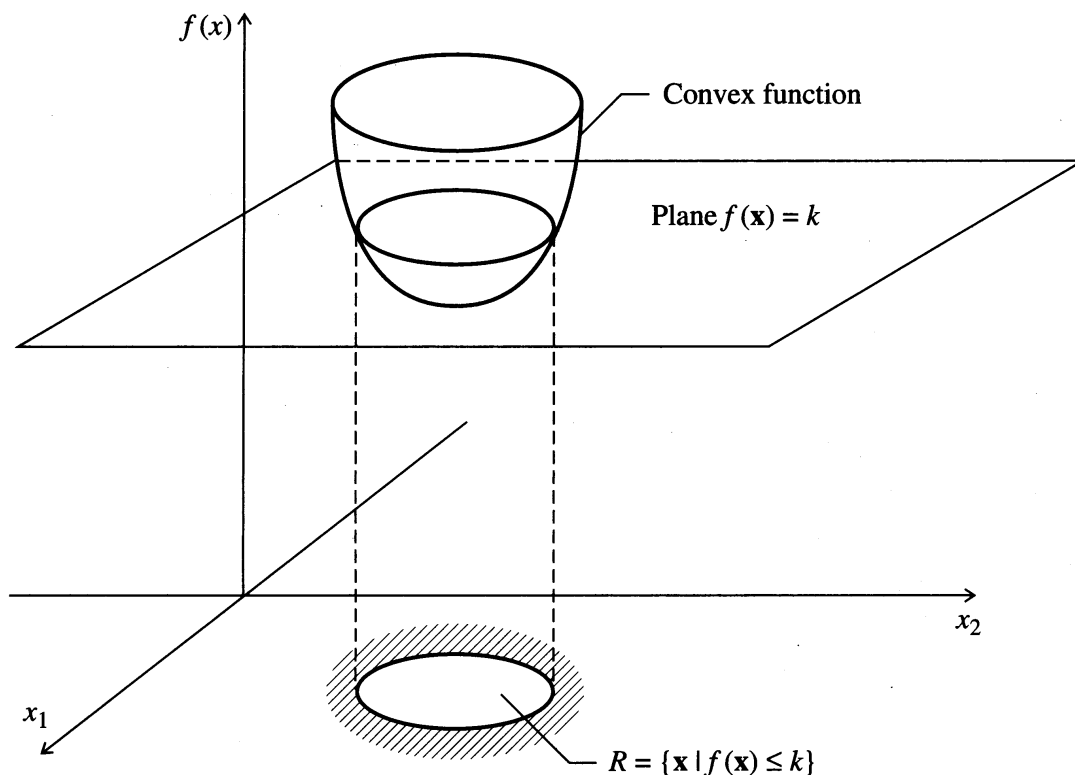
**FIGURE 4.11**

Illustration of a convex set formed by a plane $f(\mathbf{x}) = k$ cutting a convex function.

As a consequence, the problem

$$\begin{array}{ll} \text{Minimize:} & f(\mathbf{x}) \\ \text{Subject to:} & g_i(\mathbf{x}) \leq 0 \quad i = 1, \dots, m \\ & h_k(\mathbf{x}) = 0 \quad k = 1, \dots, r < n \end{array}$$

may not be a convex programming problem in the variables x_1, \dots, x_n if any of the functions $h_k(\mathbf{x})$ are nonlinear. This, of course, does not preclude efficient solution of such problems, but it does make it more difficult to guarantee the absence of local optima and to generate sharp theoretical results.

In many cases the equality constraints may be used to eliminate some of the variables, leaving a problem with only inequality constraints and fewer variables. Even if the equalities are difficult to solve analytically, it may still be worthwhile solving them numerically. This is the approach taken by the generalized reduced gradient method, which is described in Section 8.7.

Although convexity is desirable, many real-world problems turn out to be nonconvex. In addition, there is no simple way to demonstrate that a nonlinear problem is a convex problem for all feasible points. Why, then, is convex programming studied? The main reasons are

1. When convexity is assumed, many significant mathematical results have been derived in the field of mathematical programming.
2. Often results obtained under assumptions of convexity can give insight into the properties of more general problems. Sometimes, such results may even be carried over to nonconvex problems, but in a weaker form.

For example, it is usually impossible to prove that a given algorithm will find the global minimum of a nonlinear programming problem unless the problem is convex. For nonconvex problems, however, many such algorithms find at least a local minimum. Convexity thus plays a role much like that of linearity in the study of dynamic systems. For example, many results derived from linear theory are used in the design of nonlinear control systems.

Determination of convexity and concavity

The definitions of convexity and a convex function are not directly useful in establishing whether a region or a function is convex because the relations must be applied to an unbounded set of points. The following is a helpful property arising from the concept of a convex set of points. A set of points \mathbf{x} satisfying the relation

$$\mathbf{x}^T \mathbf{H}(\mathbf{x}) \mathbf{x} \leq 1$$

is convex if the Hessian matrix $\mathbf{H}(\mathbf{x})$ is a real symmetric positive-semidefinite matrix. $\mathbf{H}(\mathbf{x})$ is another symbol for $\nabla^2 f(\mathbf{x})$, the matrix of second partial derivative of $f(\mathbf{x})$ with respect to each x_i

$$\mathbf{H}(\mathbf{x}) \equiv \mathbf{H} \equiv \nabla^2 f(\mathbf{x})$$

The status of \mathbf{H} can be used to identify the character of extrema. A quadratic form $Q(\mathbf{x}) = \mathbf{x}^T \mathbf{H} \mathbf{x}$ is said to be *positive-definite* if $Q(\mathbf{x}) > 0$ for all $\mathbf{x} \neq \mathbf{0}$, and said to be *positive-semidefinite* if $Q(\mathbf{x}) \geq 0$ for all $\mathbf{x} \neq \mathbf{0}$. *Negative-definite* and *negative-semidefinite* are analogous except the inequality sign is reversed. If $Q(\mathbf{x})$ is positive-definite (semidefinite), $\mathbf{H}(\mathbf{x})$ is said to be a positive-definite (semidefinite) matrix. These concepts can be summarized as follows:

1. \mathbf{H} is *positive-definite* if and only if $\mathbf{x}^T \mathbf{H} \mathbf{x}$ is > 0 for all $\mathbf{x} \neq \mathbf{0}$.
2. \mathbf{H} is *negative-definite* if and only if $\mathbf{x}^T \mathbf{H} \mathbf{x}$ is < 0 for all $\mathbf{x} \neq \mathbf{0}$.
3. \mathbf{H} is *positive-semidefinite* if and only if $\mathbf{x}^T \mathbf{H} \mathbf{x}$ is ≥ 0 for all $\mathbf{x} \neq \mathbf{0}$.
4. \mathbf{H} is *negative-semidefinite* if and only if $\mathbf{x}^T \mathbf{H} \mathbf{x}$ is ≤ 0 for all $\mathbf{x} \neq \mathbf{0}$.
5. \mathbf{H} is *indefinite* if $\mathbf{x}^T \mathbf{H} \mathbf{x} < 0$ for some \mathbf{x} and > 0 for other \mathbf{x} .

It can be shown from a Taylor series expansion that if $f(\mathbf{x})$ has continuous second partial derivatives, $f(\mathbf{x})$ is concave if and only if its Hessian matrix is negative-semidefinite. For $f(\mathbf{x})$ to be strictly concave, \mathbf{H} must be negative-definite. For $f(\mathbf{x})$ to be convex $\mathbf{H}(\mathbf{x})$ must be positive-semidefinite and for $f(\mathbf{x})$ to be strictly convex, $\mathbf{H}(\mathbf{x})$ must be positive-definite.

EXAMPLE 4.3 ANALYSIS FOR CONVEXITY AND CONCAVITY

For each of these functions

- (a) $f(x) = 3x^2$
- (b) $f(x) = 2x$
- (c) $f(x) = -5x^2$
- (d) $f(x) = 2x^2 - x^3$

determine if $f(x)$ is convex, concave, strictly convex, strictly concave, all, or none of these classes in the range $-\infty \leq x \leq \infty$.

Solution

- (a) $f''(x) = 6$, always positive, hence $f(x)$ is both strictly convex and convex.
 (b) $f''(x) = 0$ for all values of x , hence $f(x)$ is convex and concave. Note straight lines are both convex and concave simultaneously.
 (c) $f''(x) = -10$, always negative, hence $f(x)$ is both strictly concave and concave.
 (d) $f''(x) = 6 - 3x$; may be positive or negative depending on the value of x , hence $f(x)$ is not convex or concave over the entire range of x .

For a multivariate function, the nature of convexity can best be evaluated by examining the eigenvalues of $f(\mathbf{x})$ as shown in Table 4.1 We have omitted the indefinite case for \mathbf{H} , that is when $f(\mathbf{x})$ is neither convex or concave.

TABLE 4.1
Relationship between the character of $f(\mathbf{x})$ and the state of $\mathbf{H}(\mathbf{x})$

$f(\mathbf{x})$ is	$\mathbf{H}(\mathbf{x})$ is	All the eigenvalues of $\mathbf{H}(\mathbf{x})$ are
Strictly convex	Positive-definite	>0
Convex	Positive-semidefinite	≥ 0
Concave	Negative-semidefinite	≤ 0
Strictly concave	Negative-definite	<0

Now let us further illustrate the ideas presented in this section by some examples.

EXAMPLE 4.4 DETERMINATION OF POSITIVE-DEFINITENESS OF A FUNCTION

Classify the function $f(\mathbf{x}) = 2x_1^2 - 3x_1x_2 + 2x_2^2$ using the categories in Table 4.1, or state that it does not belong in any of the categories.

Solution

$$\frac{\partial f(x)}{\partial x_1} = 4x_1 - 3x_2 \quad \frac{\partial^2 f(x)}{\partial x_1^2} = 4 \quad \frac{\partial^2 f(x)}{\partial x_2^2} = 4$$

$$\frac{\partial f(x)}{\partial x_2} = -3x_1 + 4x_2 \quad \frac{\partial^2 f(x)}{\partial x_1 \partial x_2} = \frac{\partial^2 f(x)}{\partial x_2 \partial x_1} = -3 \quad \mathbf{H}(\mathbf{x}) = \begin{bmatrix} 4 & -3 \\ -3 & 4 \end{bmatrix}$$

The eigenvalues of \mathbf{H} are 7 and 1, hence $\mathbf{H}(\mathbf{x})$ is positive-definite. Consequently, $f(\mathbf{x})$ is strictly convex (as well as convex).

EXAMPLE 4.5 DETERMINATION OF POSITIVE-DEFINITENESS OF A FUNCTION

Repeat the analysis of Example 4.4 for $f(\mathbf{x}) = x_1^2 + x_1x_2 + 2x_2 + 4$

Solution

$$\mathbf{H}(\mathbf{x}) = \begin{bmatrix} 2 & 1 \\ 1 & 0 \end{bmatrix}$$

The eigenvalues are $1 + \sqrt{2}$ and $1 - \sqrt{2}$, or one positive or one negative value. Consequently, $f(\mathbf{x})$ does not fall into any of the categories in Table 4.1. We conclude that no unique extremum exists.

EXAMPLE 4.6 DETERMINATION OF CONVEXITY AND CONCAVITY

Determine if the following function

$$f(\mathbf{x}) = 2x_1 + 3x_2 + 6$$

is convex or concave.

Solution

$$\mathbf{H}(\mathbf{x}) = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

hence the function is both convex and concave.

EXAMPLE 4.7 DETERMINATION OF CONVEXITY OF A FUNCTION

Consider the following objective function: Is it convex?

$$f(\mathbf{x}) = 2x_1^2 + 2x_1x_2 + 1.5x_2^2 + 7x_1 + 8x_2 + 24$$

Solution

$$\frac{\partial^2 f(x)}{\partial x_1^2} = 4 \quad \frac{\partial^2 f(x)}{\partial x_2^2} = 3 \quad \frac{\partial^2 f(x)}{\partial x_1 \partial x_2} = \frac{\partial^2 f(x)}{\partial x_2 \partial x_1} = 2$$

Therefore the Hessian matrix is

$$\mathbf{H}(\mathbf{x}) = \begin{bmatrix} 4 & 2 \\ 2 & 3 \end{bmatrix}$$

The eigenvalues of $\mathbf{H}(\mathbf{x})$ are 5.56 and 1.44. Because both eigenvalues are positive, the function is strictly convex (and convex, of course) for all values of x_1 and x_2 .

EXAMPLE 4.8 DETECTION OF A CONVEX REGION

Does the following set of constraints that form a closed region form a convex region?

$$-x_1^2 + x_2 \geq 1$$

$$x_1 - x_2 \geq -2$$

Solution. A plot of the two functions indicates that the region circumscribed is closed. The arrows in Figure E4.8 designate the directions in which the inequalities hold. Write the inequality constraints as $g_i \geq 0$. Therefore

$$g_1(\mathbf{x}) = -x_1^2 + x_2 - 1 \geq 0$$

$$g_2(\mathbf{x}) = x_1 - x_2 + 2 \geq 0$$

That the enclosed region is convex can be demonstrated by showing that both $g_1(\mathbf{x})$ and $g_2(\mathbf{x})$ are concave functions:

$$\mathbf{H}[g_1(\mathbf{x})] = \begin{bmatrix} -2 & 0 \\ 0 & 0 \end{bmatrix} \quad \text{negative definite}$$

$$\mathbf{H}[g_2(\mathbf{x})] = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \quad \text{negative semidefinite}$$

Because all eigenvalues are zero or negative, according to Table 4.1 both g_1 and g_2 are concave and the region is convex.

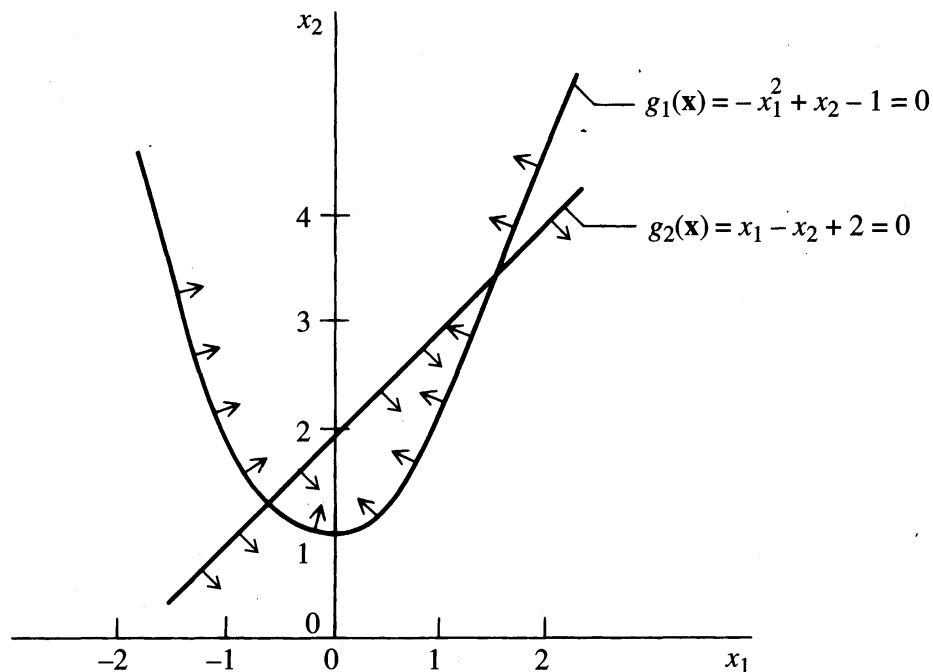


FIGURE E4.8

Convex region composed of two concave functions.

EXAMPLE 4.9 CONSTRUCTION OF A CONVEX REGION

Construct the region given by the following inequality constraints; is it convex?

$$x_1 \leq 6; x_2 \leq 6; x_1 \geq 0; x_1 + x_2 \leq 6; x_2 \geq 0$$

Solution. See Figure E4.9 for the region delineated by the inequality constraints. By visual inspection, the region is convex. This set of linear inequality constraints forms a convex region because all the constraints are concave. In this case the convex region is closed.

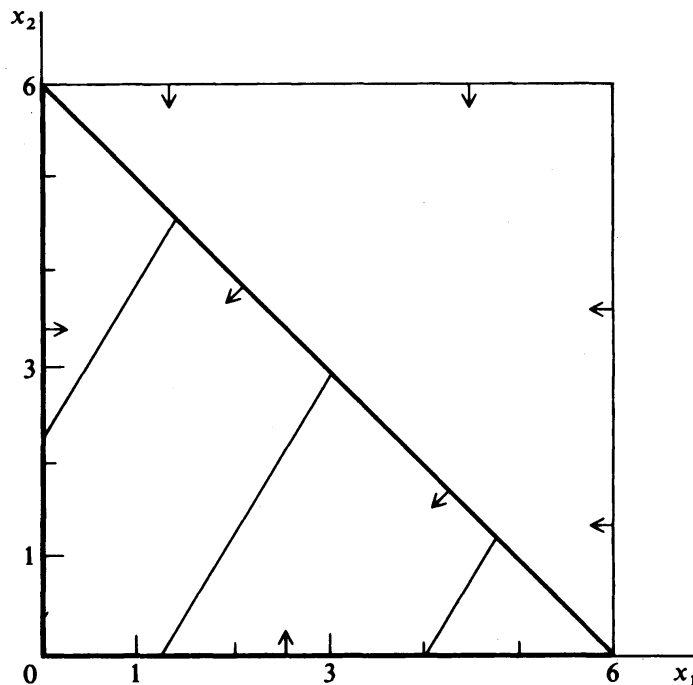


FIGURE E4.9

Diagram of region defined by linear inequality constraints.

4.4 INTERPRETATION OF THE OBJECTIVE FUNCTION IN TERMS OF ITS QUADRATIC APPROXIMATION

If a function of two variables is quadratic or approximated by a quadratic function $f(\mathbf{x}) = b_0 + b_1x_1 + b_2x_2 + b_{11}x_1^2 + b_{22}x_2^2 + b_{12}x_1x_2$, then the eigenvalues of $\mathbf{H}(\mathbf{x})$ can be calculated and used to interpret the nature of $f(\mathbf{x})$ at \mathbf{x}^* . Table 4.2 lists some conclusions that can be reached by examining the eigenvalues of $\mathbf{H}(\mathbf{x})$ for a function of two variables, and Figures 4.12 through 4.15 illustrate the different types of surfaces corresponding to each case that arises for quadratic function. By

TABLE 4.2
Geometric interpretation of a quadratic function

Case	Eigenvalue relations	Signs		Types of contours	Geometric interpretation	Character of center of contours	Figure
		e_1	e_2				
1	$e_1 = e_2$	-	-	Circles	Circular hill	Maximum	4.12
2	$e_1 = e_2$	+	+	Circles	Circular valley	Minimum	4.12
3	$e_1 > e_2$	-	-	Ellipses	Elliptical hill	Maximum	4.12
4	$e_1 > e_2$	+	+	Ellipses	Elliptical valley	Minimum	4.12
5	$ e_1 = e_2 $	+	-	Hyperbolas	Symmetrical saddle	Saddle point	4.13
6	$ e_1 = e_2 $	-	+	Hyperbolas	Symmetrical saddle	Saddle point	4.13
7	$e_1 > e_2$	+	-	Hyperbolas	Elongated saddle	Saddle point	4.13
8	$e_2 = 0$	-		Straight lines	Stationary ridge*	None	4.14
9	$e_2 = 0$	+		Straight lines	Stationary valley*	None	4.14
10	$e_2 = 0$	-		Parabolas	Rising ridge*‡	At ∞	4.15
11	$e_2 = 0$	+		Parabolas	Falling valley*‡	At ∞	4.15

*These are "degenerate" surfaces.

‡The condition of rising or falling must be evaluated from the linear terms in $f(x)$.

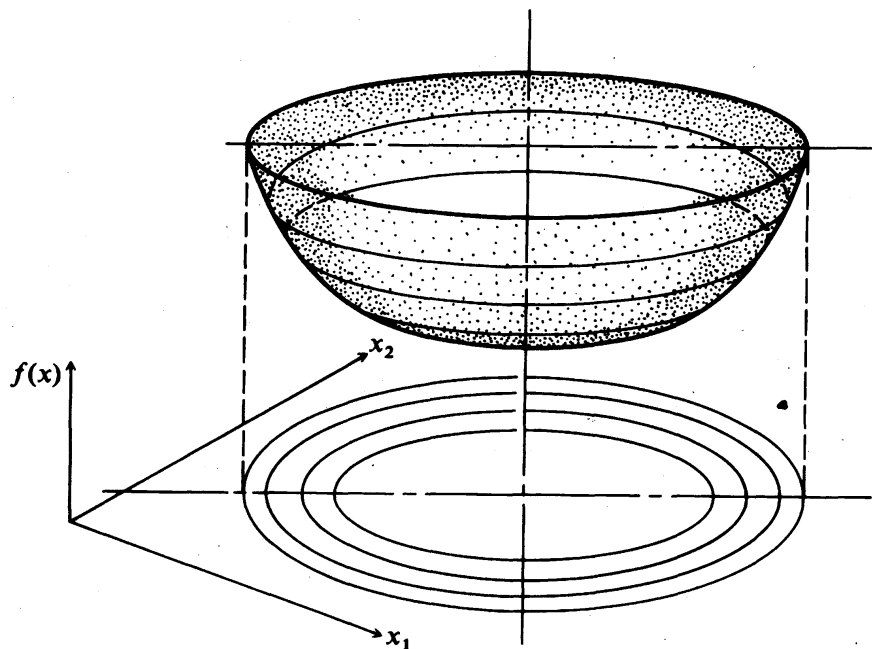


FIGURE 4.12

Geometry of a quadratic objective function of two independent variables—elliptical contours. If the eigenvalues are equal, then the contours are circles.

implication, analysis of a function of many variables via examination of the eigenvalues can be conducted, whereas contour plots are limited to functions of only two or three variables.

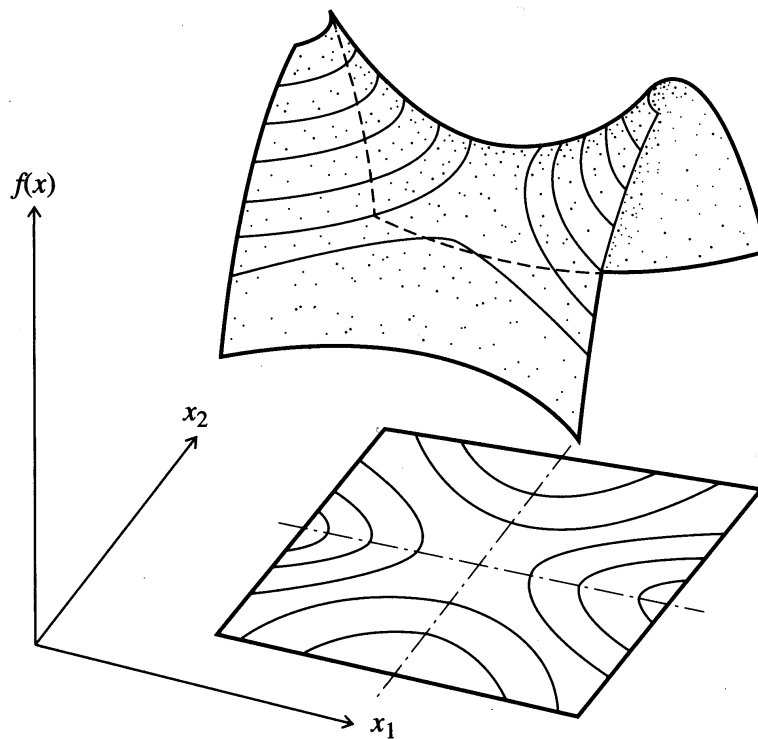


FIGURE 4.13
Geometry of a quadratic objective function of two independent variables—saddle point.

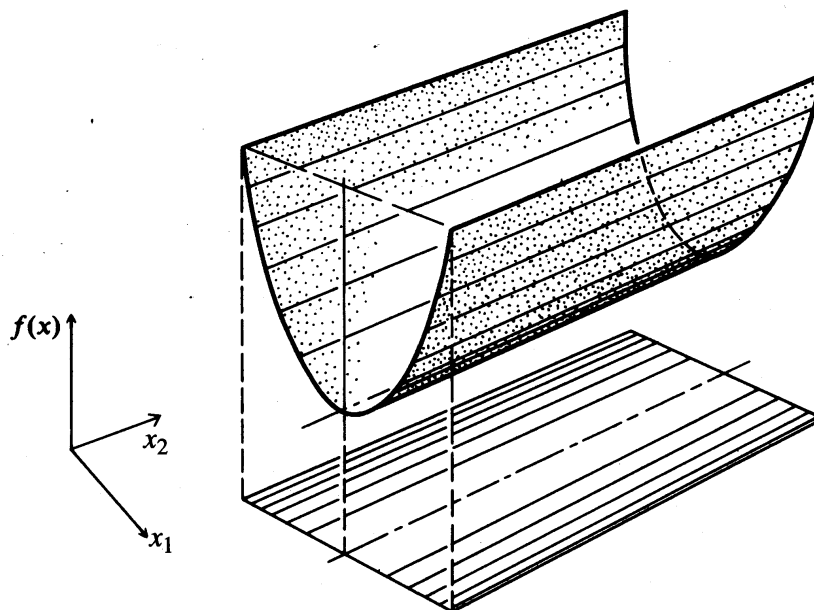
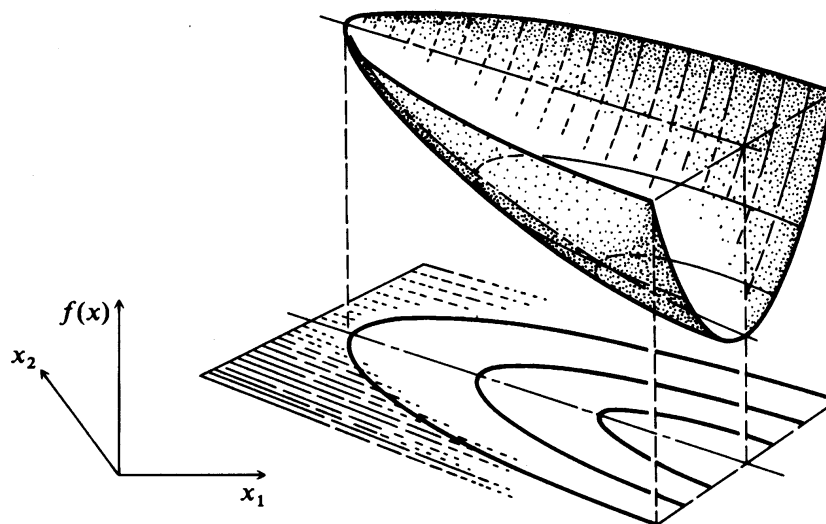


FIGURE 4.14
Geometry of a quadratic objective function of two independent variables—stationary valley.

**FIGURE 4.15**

Geometry of second-order objective function of two independent variables—falling valley.

Figure 4.12 corresponds to objective functions in well-posed optimization problems. In Table 4.2, cases 1 and 2 correspond to contours of $f(\mathbf{x})$ that are concentric circles, but such functions rarely occur in practice. Elliptical contours such as correspond to cases 3 and 4 are most likely for well-behaved functions. Cases 5 to 10 correspond to degenerate problems, those in which no finite maximum or minimum or perhaps nonunique optima appear.

For well-posed quadratic objective functions the contours always form a convex region; for more general nonlinear functions, they do not (see the next section for an example). It is helpful to construct contour plots to assist in analyzing the performance of multivariable optimization techniques when applied to problems of two or three dimensions. Most computer libraries have contour plotting routines to generate the desired figures.

As indicated in Table 4.2, the eigenvalues of the Hessian matrix of $f(\mathbf{x})$ indicate the shape of a function. For a positive-definite symmetric matrix, the eigenvectors (refer to Appendix A) form an orthonormal set. For example, in two dimensions, if the eigenvectors are \mathbf{v}_1 and \mathbf{v}_2 , $\mathbf{v}_1^T \mathbf{v}_2 = 0$ (the eigenvectors are perpendicular to each other). The eigenvectors also correspond to the directions of the principal axes of the contours of $f(\mathbf{x})$.

One of the primary requirements of any successful optimization technique is the ability to move rapidly in a local region along a narrow valley (in minimization) toward the minimum of the objective function. In other words, an efficient algorithm selects a search direction that generally follows the axis of the valley rather than jumping back and forth across the valley. Valleys (ridges in maximization) occur quite frequently, at least locally, and these types of surfaces have the potential to slow down greatly the search for the optimum. A valley lies in the direction of the eigenvector associated with a small eigenvalue of the Hessian

matrix of the objective function. For example, if the Hessian matrix of a quadratic function is

$$\mathbf{H} = \begin{bmatrix} 1 & 0 \\ 0 & 10 \end{bmatrix}$$

then the eigenvalues are $e_1 = 1$ and $e_2 = 10$. The eigenvector associated with $e_1 = 1$, that is, the x_1 axis, is lined up with the valley in the ellipsoid. Variable transformation techniques can be used to allow the problem to be more efficiently solved by a search technique (see Chapter 6).

Valleys and ridges corresponding to cases 1 through 4 can lead to a minimum or maximum, respectively, but not for cases 8 through 11. Do you see why?

4.5 NECESSARY AND SUFFICIENT CONDITIONS FOR AN EXTREMUM OF AN UNCONSTRAINED FUNCTION

Figure 4.16 illustrates the character of $f(x)$ if the objective function is a function of a single variable. Usually we are concerned with finding the minimum or maximum of a multivariable function $f(\mathbf{x})$. The problem can be interpreted geometrically as finding the point in an n -dimension space at which the function has an extremum. Examine Figure 4.17 in which the contours of a function of two variables are displayed.

An optimal point \mathbf{x}^* is completely specified by satisfying what are called the *necessary* and *sufficient conditions* for optimality. A condition N is necessary for a result R if R can be true only if the condition is true ($R \Rightarrow N$). The reverse is not true, however, that is, if N is true, R is not necessarily true. A condition is sufficient for a result R if R is true if the condition is true ($S \Rightarrow R$). A condition T is necessary and sufficient for result R if R is true if and only if T is true ($T \Leftrightarrow R$).

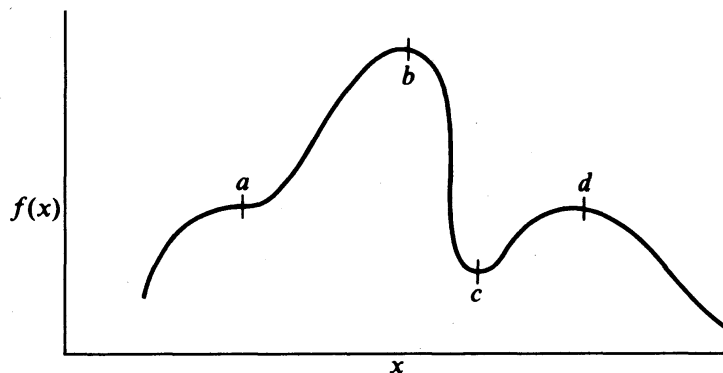
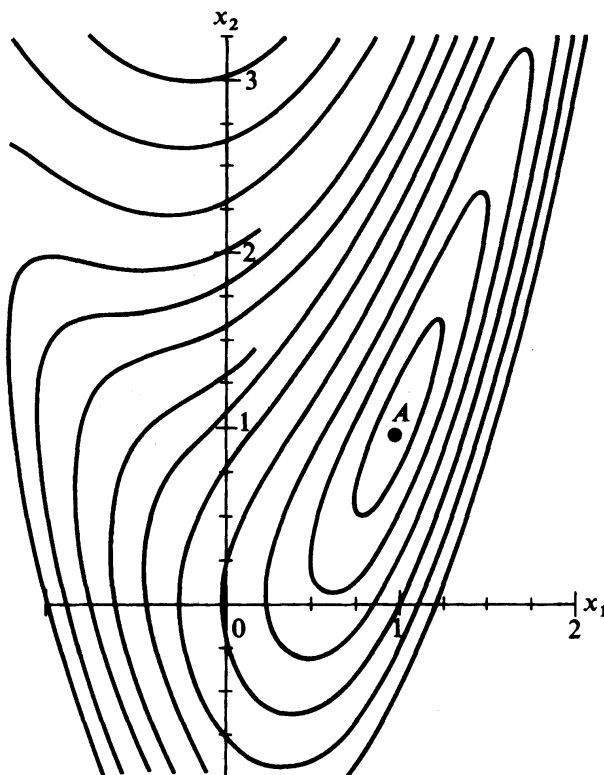


FIGURE 4.16

A function exhibiting different types of stationary points.

Key: a —inflection point (scalar equivalent to a saddle point);
 b —global maximum (and local maximum); c —local minimum;
 d —local maximum

**FIGURE 4.17a**

A function of two variables with a single stationary point (the extremum).

The easiest way to develop the necessary and sufficient conditions for a minimum or maximum of $f(\mathbf{x})$ is to start with a Taylor series expansion about the presumed extremum \mathbf{x}^*

$$f(\mathbf{x}) = f(\mathbf{x}^*) + \nabla^T f(\mathbf{x}^*) \Delta \mathbf{x} + \frac{1}{2} (\Delta \mathbf{x}^T) \nabla^2 f(\mathbf{x}^*) \Delta \mathbf{x} + O_3(\Delta \mathbf{x}) + \dots \quad (4.4)$$

where $\Delta \mathbf{x} = \mathbf{x} - \mathbf{x}^*$, the perturbation of \mathbf{x} from \mathbf{x}^* . We assume all terms in Equation (4.4) exist and are continuous, but will ignore the terms of order 3 or higher [$O_3(\Delta \mathbf{x})$], and simply analyze what occurs for various cases involving just the terms through the second order.

We defined a local minimum as a point \mathbf{x}^* such that no other point in the vicinity of \mathbf{x}^* yields a value of $f(\mathbf{x})$ less than $f(\mathbf{x}^*)$, or

$$f(\mathbf{x}) - f(\mathbf{x}^*) \geq 0 \quad (4.5)$$

\mathbf{x}^* is a global minimum if Equation (4.5) holds for any \mathbf{x} in the n -dimensional space of \mathbf{x} . Similarly, \mathbf{x}^* is a local maximum if

$$f(\mathbf{x}) - f(\mathbf{x}^*) \leq 0 \quad (4.6)$$

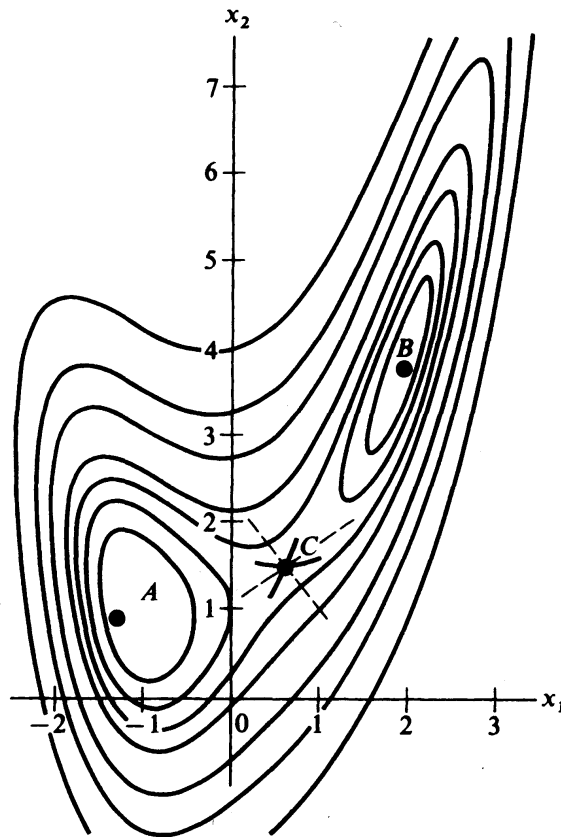


FIGURE 4.17b

A function of two variables with three stationary points and two extrema, A and B .

Examine the second term on the right-hand side of Equation (4.4): $\nabla^T f(\mathbf{x}^*) \Delta \mathbf{x}$. Because $\Delta \mathbf{x}$ is arbitrary and can have both plus and minus values for its elements, we must insist that $\nabla f(\mathbf{x}^*) = \mathbf{0}$. Otherwise the resulting term added to $f(\mathbf{x}^*)$ would violate Equation (4.5) for a minimum, or Equation (4.6) for a maximum. Hence, a *necessary condition* for a minimum or maximum of $f(\mathbf{x})$ is that the gradient of $f(\mathbf{x})$ vanishes at \mathbf{x}^*

$$\nabla f(\mathbf{x}^*) = \mathbf{0} \quad (4.7)$$

that is, \mathbf{x}^* is a stationary point.

With the second term on the right-hand side of Equation (4.4) forced to be zero, we next examine the third term: $\frac{1}{2}(\Delta \mathbf{x}^T) \nabla^2 f(\mathbf{x}^*) \Delta \mathbf{x}$. This term establishes the character of the stationary point (minimum, maximum, or saddle point). In Figure 4.17b, A and B are minima and C is a saddle point. Note how movement along one of the perpendicular search directions (dashed lines) from point C increases $f(\mathbf{x})$, whereas movement in the other direction decreases $f(\mathbf{x})$. Thus, satisfaction of the necessary conditions does not guarantee a minimum or maximum.

To establish the existence of a minimum or maximum at \mathbf{x}^* , we know from Equation (4.4) with $\nabla f(\mathbf{x}^*) = \mathbf{0}$ and the conclusions reached in Section 4.3 concerning convexity that for $\Delta \mathbf{x} \neq \mathbf{0}$ we have the following outcomes

$\nabla^2 f(\mathbf{x}^*) \equiv \mathbf{H}(\mathbf{x}^*)$	$\Delta \mathbf{x}^T \nabla^2 f(\mathbf{x}^*) \Delta \mathbf{x}$	Near \mathbf{x}^* , $f(\mathbf{x}) - f(\mathbf{x}^*)$
Positive-definite	>0	Increases
Positive-semidefinite	≥ 0	Possibly increases
Negative-definite	<0	Decreases
Negative-semidefinite	≤ 0	Possibly decreases
Indefinite	Both ≤ 0 and ≥ 0 depending on $\Delta \mathbf{x}$	Increases, decreases, neither

Consequently, \mathbf{x}^* can be classified as

$\nabla^2 f(\mathbf{x}^*) \equiv \mathbf{H}(\mathbf{x}^*)$	\mathbf{x}^*
Positive-definite	Unique ("isolated") minimum
Negative-definite	Unique ("isolated") maximum

These two conditions are known as the *sufficiency conditions*.

In summary, the necessary conditions (items 1 and 2 in the following list) and the sufficient condition (3) to guarantee that \mathbf{x}^* is an extremum are as follows:

1. $f(\mathbf{x})$ is twice differentiable at \mathbf{x}^* .
2. $\nabla f(\mathbf{x}^*) = \mathbf{0}$, that is, a stationary point exists at \mathbf{x}^* .
3. $\mathbf{H}(\mathbf{x}^*)$ is positive-definite for a minimum to exist at \mathbf{x}^* , and negative-definite for a maximum to exist at \mathbf{x}^* .

Of course, a minimum or maximum may exist at \mathbf{x}^* even though it is not possible to demonstrate the fact using the three conditions. For example, if $f(x) = x^{4/3}$, $x^* = 0$ is a minimum but $\mathbf{H}(0)$ is not defined at $x^* = 0$, hence condition 3 is not satisfied.

EXAMPLE 4.10 CALCULATION OF A MINIMUM OF $f(x)$

Does $f(x) = x^4$ have an extremum? If so, what is the value of x^* and $f(x^*)$ at the extremum?

Solution

$$f'(x) = 4x^3 \quad f''(x) = 12x^2$$

Set $f'(x) = 0$ and solve for x ; hence $x = 0$ is a stationary point. Also, $f''(0) = 0$, meaning that condition 3 is *not* satisfied. Figure E4.10 is a plot of $f(x) = x^4$. Thus, a minimum exists for $f(x)$ but the sufficiency condition is not satisfied.

If both first and second derivatives vanish at the stationary point, then further analysis is required to evaluate the nature of the function. For functions of a single variable, take successively higher derivatives and evaluate them at the stationary point. Continue this procedure until one of the higher derivatives is not zero (the n th one); hence, $f'(x^*)$, $f''(x^*)$, \dots , $f^{(n-1)}(x^*)$ all vanish. Two cases must be analyzed:

1. If n is even, the function attains a maximum or a minimum; a positive sign of $f^{(n)}$ indicates a minimum, a negative sign a maximum.
2. If n is odd, the function exhibits a saddle point.

For more details refer to Beveridge and Schechter (1970).

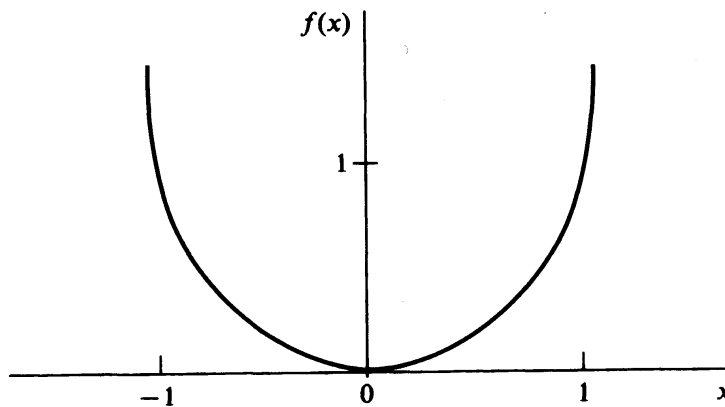


FIGURE E4.10

For application of these guidelines to $f(x) = x^4$, you will find $d^4f(x)/dx^4 = 24$ for which n is even and the derivative is positive, so that a minimum exists.

EXAMPLE 4.11 CALCULATION OF EXTREMA

Identify the stationary points of the following function (Fox, 1971), and determine if any extrema exist.

$$f(\mathbf{x}) = 4 + 4.5x_1 - 4x_2 + x_1^2 + 2x_2^2 - 2x_1x_2 + x_1^4 - 2x_1^2x_2$$

Solution. For this function, three stationary points can be located by setting $\nabla f(\mathbf{x}) = \mathbf{0}$:

$$\frac{\partial f(x)}{\partial x_1} = 4.5 + 2x_1 - 2x_2 + 4x_1^3 - 4x_1x_2 = 0 \quad (a)$$

$$\frac{\partial f(x)}{\partial x_2} = -4 + 4x_2 - 2x_1 - 2x_1^2 = 0 \quad (b)$$

The set of nonlinear equations (a) and (b) has to be solved, say by Newton's method, to get the pairs (x_1, x_2) as follows:

Point	Stationary point (x_1, x_2)	$f(\mathbf{x})$	Hessian matrix eigenvalues		Classification
B	(1.941, 3.854)	0.9855	37.03	0.97	Local minimum
A	(-1.053, 1.028)	-0.5134	10.5	3.5	Local minimum (also the global minimum)
C	(0.6117, 1.4929)	2.83	7.0	-2.56	Saddle point

Figure 4.17b shows contours for the objective function in this example. Note that the global minimum can only be identified by evaluating $f(\mathbf{x})$ for all the local minima.

For general nonlinear objective functions, it is usually difficult to ascertain the nature of the stationary points without detailed examination of each point.

EXAMPLE 4.12

In many types of processes such as batch constant-pressure filtration or fixed-bed ion exchange, the production rate decreases as a function of time. At some optimal time t^{opt} , production is terminated (at P^{opt}) and the equipment is cleaned. Figure E4.12a illustrates the cumulative throughput $P(t)$ as a function of time t for such a process. For one cycle of production and cleaning, the overall production rate is

$$R(t) = \frac{P(t)}{t + t_c} \quad (a)$$

where $R(t)$ = the overall production rate per cycle (mass/time)
 t_c = the cleaning time (assumed to be constant)

Determine the maximum production rate and show that P^{opt} is indeed the maximum throughput.

Solution. Differentiate $R(t)$ with respect to t , and equate the derivative to 0:

$$\frac{dR(t)}{dt} = \frac{-P(t) + [dP(t)/dt](t + t_c)}{(t + t_c)^2} = 0$$

$$P^{\text{opt}} = \left. \frac{dP(t)}{dt} \right|_{\text{opt}} (t + t_c) \quad (b)$$

The geometric interpretation of Equation (b) is the classical result (Walker et al., 1937) that the tangent to $P(t)$ at P^{opt} intersects the time axis at $-t_c$. Examine Figure E4.12b. The maximum overall production rate is

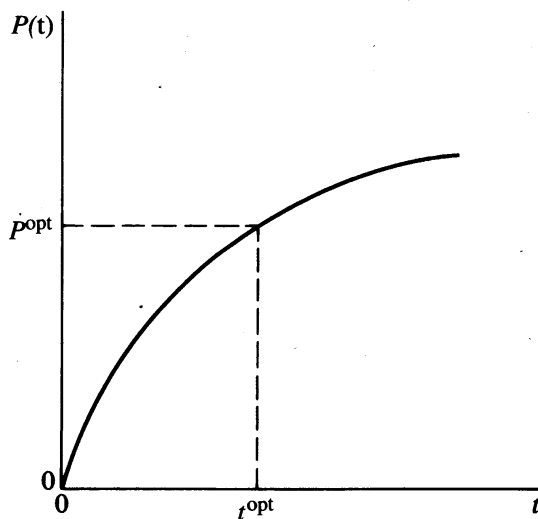


FIGURE E4.12a

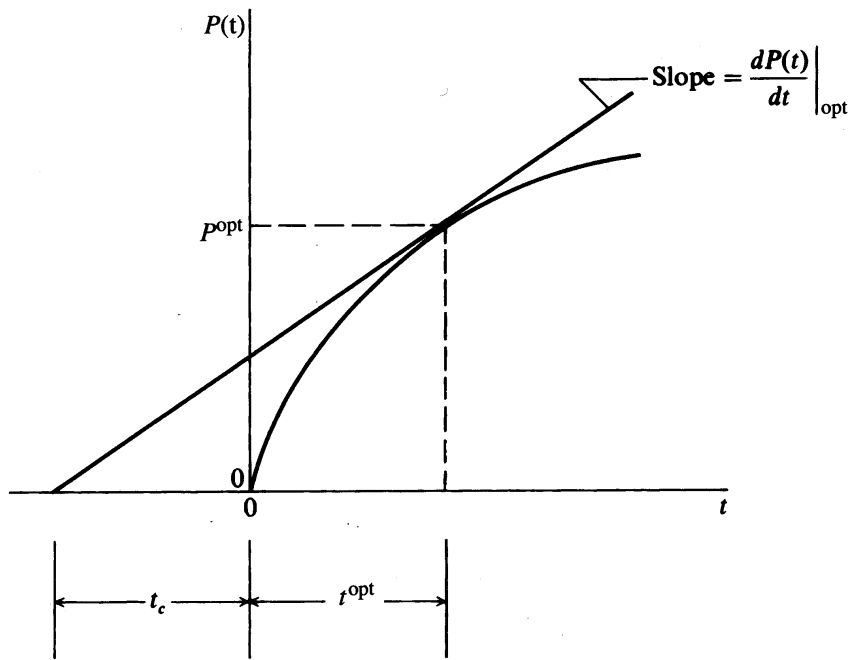


FIGURE E4.12b

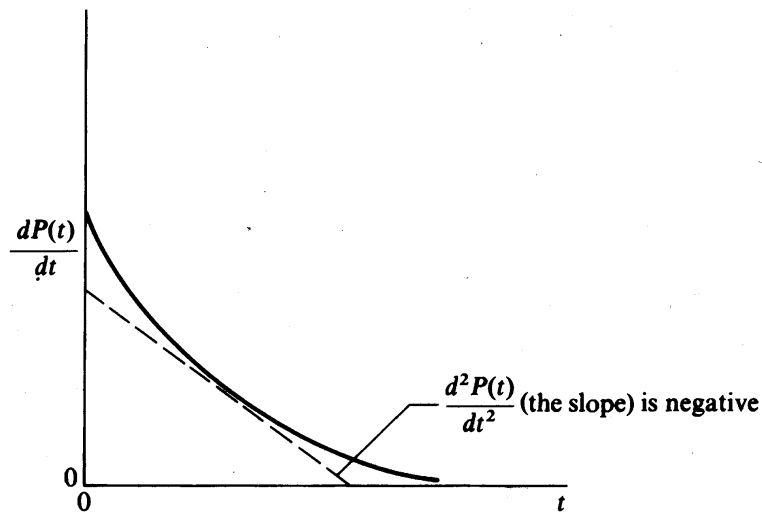


FIGURE E4.12c

$$R^{\text{opt}} = \frac{P^{\text{opt}}}{t^{\text{opt}} + t_c} \quad (c)$$

Does P^{opt} meet the sufficiency condition to be a maximum? Is

$$\frac{d^2R(t)}{dt^2} = \frac{2P(t) - 2[dP(t)/dt](t + t_c) + [d^2P(t)/dt^2](t + t_c)^2}{(t + t_c)^3} < 0 \quad ? \quad (d)$$

Rearrangement of (d) and introduction of (b) into (d), or the pair $(P^{\text{opt}}, t^{\text{opt}})$, gives

$$\frac{d^2P(t)}{dt^2}(t + t_c)^2 < 0$$

From Figure E4.12b we note in the range $0 < t < t^{\text{opt}}$ that $dP(t)/dt$ is always positive and decreasing so that $d^2P(t)/dt^2$ is always negative (see Figure E4.12c). Consequently, the sufficiency condition is met.

REFERENCES

- Bartela, R.; J. Beatty; and B. Barsky. *An Introduction to Splines for Use in Computer Graphics and Geometric Modeling*. Morgan Kaufman Publishers, Los Altos, CA (1987).
- Beveridge, G. S. G.; and R. S. Schechter. *Optimization: Theory and Practice*, McGraw-Hill, New York (1970), p. 126.
- Fox, R. L. *Optimization Methods for Engineering Design*. Addison-Wesley, Reading, MA (1971), p. 42.
- Happell, J.; and D. G. Jordan. *Chemical Process Economics*. Marcel Dekker, New York (1975), p. 178.
- Kaplan, W. *Maxima and Minima with Applications: Practical Optimization and Duality*. Wiley, New York (1998).
- Nash, S. G.; and A. Sofer. *Linear and Nonlinear Programming*. McGraw-Hill, New York (1996).
- Noltie, C. B. *Optimum Pipe Size Selection*. Gulf Publ. Co., Houston, Texas (1978).
- Walker, W. H.; W. K. Lewis; W. H. McAdams; and E. R. Gilliland. *Principles of Chemical Engineering*. 3d ed. McGraw-Hill, New York (1937), p. 357.

SUPPLEMENTARY REFERENCES

- Amundson, N. R. *Mathematical Methods in Chemical Engineering: Matrices and Their Application*. Prentice-Hall, Englewood Cliffs, New Jersey (1966).
- Avriel, M. *Nonlinear Programming*. Prentice-Hall, Englewood Cliffs, New Jersey (1976).
- Bazarra, M. S.; H. D. Sherali; and C. M. Shetty. *Nonlinear Programming: Theory and Algorithms*. Wiley, New York (1993).
- Borwein, J.; and A. S. Lewis. *Convex Analysis and Nonlinear Optimization*. Springer, New York (1999).
- Jeter, M. W. *Mathematical Programming*. Marcel Dekker, New York (1986).
- Luenberger, D. G. *Linear and Nonlinear Programming*. 2nd ed. Addison-Wesley, Menlo Park, CA (1984).

PROBLEMS

4.1 Classify the following functions as continuous (specify the range) or discrete:

(a) $f(x) = e^x$

(b) $f(x) = ax_{n-1} + b(x_0 - x_n)$ where x_n represents a stage in a distillation column

(c) $f(x) = \frac{x_D - x_s}{1 + x_s}$ where x_D = concentration of vapor from a still and x_s is the concentration in the still

4.2 The future worth S of a series of n uniform payments each of amount P is

$$S = \frac{P}{i}[(1 + i)^n - 1]$$

where i is the interest rate per period. If i is considered to be the only variable, is it discrete or continuous? Explain. Repeat for n . Repeat for both n and i being variables.

4.3 In a plant the gross profit P in dollars is

$$P = nS - (nV + F)$$

where n = the number of units produced per year

S = the sales price in dollars per unit

V = the variable cost of production in dollars per unit

F = the fixed charge in dollars

Suppose that the average unit cost is calculated as

$$\text{Average unit cost} = \frac{nV + F}{n}$$

Discuss under what circumstances n can be treated as a continuous variable.

4.4 One rate of return is the ratio of net profit P to total investment

$$R = 100 \frac{P(1 - t)}{I} = 100(1 - t) \frac{[S - (V + F/n)]}{I/n}$$

where t = the fraction tax rate

I = the total investment in dollars

Find the maximum R as a function of n for a given I if n is a continuous variable.

Repeat if n is discrete. (See Problem 4.3 for other notation.)

4.5 Rewrite the following linear programming problems in matrix notation.

(a) Minimize: $f(\mathbf{x}) = 3x_1 + 2x_2 + x_3$
 Subject to: $g_1(\mathbf{x}) = 2x_1 + 3x_2 + x_3 \geq 10$
 $g_2(\mathbf{x}) = x_1 + 2x_2 + x_3 \geq 15$

(b) Maximize: $f(\mathbf{x}) = 5x_1 + 10x_2 + 12x_3$
 Subject to: $g_1(\mathbf{x}) = 15x_1 + 10x_2 + 10x_3 \leq 200$
 $g_2(\mathbf{x}) = x_1 \geq 0$
 $g_3(\mathbf{x}) = x_2 \geq 0$
 $g_4(\mathbf{x}) = x_3 \geq 0$
 $h_1(\mathbf{x}) = 10x_1 + 25x_2 + 20x_3 = 300$

4.6 Put the following nonlinear objective function into matrix notation by defining suitable matrices; $\mathbf{x} = [x_1 \ x_2]^T$.

$$f(\mathbf{x}) = 3 + 2x_1 + 3x_2 + 2x_1^2 + 2x_1x_2 + 6x_2^2$$

4.7 Sketch the objective function and constraints of the following nonlinear programming problems.

(a) Minimize: $f(\mathbf{x}) = 2x_1^2 - 2x_1x_2 + 2x_2^2 - 6x_1 + 6$

Subject to: $g_1(\mathbf{x}) = x_1 + x_2 \leq 2$

(b) Minimize: $f(\mathbf{x}) = x_1^3 - 3x_1x_2 + 4$

Subject to: $g_1(\mathbf{x}) = 5x_1 + 2x_2 \geq 18$

$h_1(\mathbf{x}) = -2x_1 + x_2^2 = 5$

(c) Minimize: $f(\mathbf{x}) = -5x_1^2 + x_2^2$

Subject to: $g_1(\mathbf{x}) = \frac{x_1^2}{x_2^2} - \frac{1}{x_2} \leq -1$

$g_2(\mathbf{x}) = x_1 \geq 0$

$g_3(\mathbf{x}) = x_2 \geq 0$

4.8 Distinguish between the local and global extrema of the following objective function.

$$f(\mathbf{x}) = 2x_1^3 + x_2^2 + x_1^2x_2^2 + 4x_1x_2 + 3$$

4.9 Are the following vectors (a) feasible or nonfeasible vectors with regard to Problem 4.5b; (b) interior or exterior vectors?

(1) $\mathbf{x} = [5 \ 2 \ 10]^T$

(2) $\mathbf{x} = [10 \ 2 \ 7.5]^T$

(3) $\mathbf{x} = [0 \ 0 \ 0]^T$

4.10 Shade the feasible region of the nonlinear programming problems of Problem 4.7. Is $\mathbf{x} = [1 \ 1]^T$ an interior, boundary, or exterior point in these problems?

4.11 What is the feasible region for \mathbf{x} given the following constraints? Sketch the feasible region for the two-dimensional problems.

(a) $h_1(\mathbf{x}) = x_1 + x_2 - 3 = 0$

$h_2(\mathbf{x}) = 2x_1 - x_2 + 1 = 0$

(b) $h_1(\mathbf{x}) = x_1^2 + x_2^2 + x_3^2 = 0$

$h_2(\mathbf{x}) = x_1 + x_2 + x_3 = 0$

(c) $g_1(\mathbf{x}) = x_1 - x_2^2 - 2 \geq 0$

$g_2(\mathbf{x}) = x_1 - x_2 + 4 \geq 0$

(d) $h_1(\mathbf{x}) = x_1^2 + x_2^2 + 3$

$g_1(\mathbf{x}) = x_1 - x_2 + 2 \geq 0$

$g_2(\mathbf{x}) = x_1 \geq 0$

$g_3(\mathbf{x}) = x_2 \geq 0$

4.12 Two solutions to the nonlinear programming problem

$$\begin{aligned} \text{Minimize: } & f(\mathbf{x}) = 7x_1 - 6x_2 + 4x_3 \\ \text{Subject to: } & h_1(\mathbf{x}) = x_1^2 + 2x_2^2 + 3x_3^2 - 1 = 0 \\ & h_2(\mathbf{x}) = 5x_1 + 5x_2 - 3x_3 - 6 = 0 \end{aligned}$$

have been reported, apparently a maximum and a minimum.

$$\mathbf{x} = \begin{bmatrix} 0.947 \\ 0.207 \\ -0.0772 \end{bmatrix} \quad \mathbf{x} = \begin{bmatrix} 0.534 \\ 0.535 \\ -0.219 \end{bmatrix}$$

$$f(\mathbf{x}) = 5.08 \quad f(\mathbf{x}) = -0.346$$

Verify that each of these \mathbf{x} vectors is feasible.

4.13 The problem

$$\begin{aligned} \text{Minimize: } & f(\mathbf{x}) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2 \\ \text{Subject to: } & x_1^2 + x_2^2 \leq 2 \end{aligned}$$

is reported to have a local minimum at the point $\mathbf{x}^* = [1 \ 1]^T$. Is this local optimum also a global optimum?

4.14 Under what circumstances is a local minimum guaranteed to be the global minimum? (Be brief.)**4.15** Are the following functions convex? Strictly convex? Why?

(a) $2x_1^2 + 2x_1x_2 + 3x_2^2 + 7x_1 + 8x_2 + 25$

What are the optimum values of x_1 and x_2 ?

(b) e^{5x}

4.16 Determine the convexity or concavity of the following objective functions:

(a) $f(x_1, x_2) = (x_1 - x_2)^2 + x_2^2$

(b) $f(x_1, x_2, x_3) = x_1^2 + x_2^2 + x_3^2$

(c) $f(x_1, x_2) = e^{x_1} + e^{x_2}$

4.17 Show that $f = e^{x_1} + e^{x_2}$ is convex. Is it also strictly convex?**4.18** Show that $f = |x|$ is convex.**4.19** Is the following region constructed by the four constraints convex? Closed?

$$x_2 \geq 1 - x_1$$

$$x_2 \leq 1 + 0.5x_1$$

$$x_1 \leq 2$$

$$x_2 \geq 0$$

4.20 Does the following set of constraints form a convex region?

$$g_1(x) = -(x_1^2 + x_2^2) + 9 \geq 0$$

$$g_2(x) = -x_1 - x_2 + 1 \geq 0$$

4.21 Consider the following problem:

$$\text{Minimize: } f(\mathbf{x}) = x_1^2 + x_2$$

$$\text{Subject to: } g_1(\mathbf{x}) = x_1^2 + x_2^2 - 9 \leq 0$$

$$g_2(\mathbf{x}) = (x_1 + x_2^2) - 1 \leq 0$$

$$g_3(\mathbf{x}) = (x_1 + x_2) - 1 \leq 0$$

Does the constraint set form a convex region? Is it closed? (*Hint*: A plot will help you decide.)

4.22 Is the following function convex, concave, neither, or both? Show your calculations.

$$f(\mathbf{x}) = \ln x_1 + \ln x_2$$

4.23 Sketch the region defined by the following inequality constraints. Is it a convex region? Is it closed?

$$x_1 + x_2 - 1 \geq 0$$

$$x_1 - x_2 + 1 \geq 0$$

$$2 - x_1 \geq 0$$

$$x_2 \geq 0$$

4.24 Does the following constraint set form a convex region (set)?

$$h(\mathbf{x}) = x_1^2 + x_2^2 - 9 = 0$$

$$g_1(\mathbf{x}) = -(x_1 + x_2^2) + 1 \geq 0$$

$$g_2(\mathbf{x}) = -(x_1 + x_2) + 1 \geq 0$$

4.25 Separable functions are those that can be expressed in the form

$$\psi(\mathbf{x}) = \sum_{i=1}^n \psi_i(x_i)$$

For example, $x_1^2 + x_2^2 + x_3^2$ is a separable function because

$$\psi(\mathbf{x}) = \sum x_i^2$$

Show that if the terms in a separable function are convex, the separable function is convex.

4.26 Is the following problem a convex programming problem?

$$\text{Minimize: } f(\mathbf{x}) = 100x_1 + \frac{200}{x_1x_2}$$

$$\text{Subject to: } 2x_2 + \frac{300}{x_1x_2} \leq 1$$

$$x_1, x_2 \geq 0$$

4.27 Classify each of the following matrices as (a) positive-definite, (b) negative-definite, (c) neither.

$$(a) \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (c) \begin{bmatrix} 0.1 & 0 \\ 3 & 1 \end{bmatrix}$$

$$(b) \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \quad (d) \begin{bmatrix} 1 & -2 \\ -2 & 1 \end{bmatrix}$$

4.28 Determine whether the following matrix is positive-definite, positive-semidefinite, negative-definite, negative-semidefinite, or none of the above. Show all calculations.

$$\mathbf{A} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 0 \end{bmatrix}$$

4.29 In designing a can to hold a specified amount of soda water, the cost function (to be minimized) for manufacturing one can is

$$f(D, h) = \pi Dh + \frac{\pi}{2}D^2$$

and the constraints are

$$\frac{\pi}{4}D^2h \geq 400$$

$$3.5 \leq D \leq 8 \quad 8 \leq h \leq 18$$

Based on the preceding problem, answer the following; as far as possible for each answer use mathematics to support your statements:

- State whether $f(D, h)$ is unimodal (one extremum) or multimodal (more than one extremum).
- State whether $f(D, h)$ is continuous or not.
- State whether $f(D, h)$ is convex, concave, or neither.
- State whether or not $f(D, h)$ alone meets the necessary and sufficient conditions for a minimum to exist.
- State whether the constraints form a convex region.

4.30 A reactor converts an organic compound to product P by heating the material in the presence of an additive A (mole fraction = x_A). The additive can be injected into the reactor, while steam can be injected into a heating coil inside the reactor to provide heat. Some conversion can be obtained by heating without addition of A , and vice versa.

The product P can be sold for \$50/lb mol. For 1 lb mol of feed, the cost of the additive (in dollars/lb mol) as a function of x_A is given by the formula, $2.0 + 10x_A + 20x_A^2$. The cost of the steam (in dollars) as a function of S is $1.0 + 0.003S + 2.0 \times 10^{-6} S^2$. ($S = \text{lb steam/lb mol feed}$). The yield equation is $y_p = 0.1 + 0.3x_A + 0.001S + 0.0001x_A S$; $y_p = \text{lb mol product } P/\text{lb mol feed}$.

(a) Formulate the profit function (basis of 1.0 lb mol feed) in terms of x_A and S .

$$f = \text{Income} - \text{Costs}$$

The constraints are:

$$0 \leq x_A \leq 1 \quad S \geq 0$$

(b) Is f a concave function? Demonstrate mathematically why it is or why it is not concave.

(c) Is the region of search convex? Why?

4.31 The objective function for the work requirement for a three-stage compressor can be expressed as (p is pressure)

$$f = \left(\frac{p_2}{p_1}\right)^{0.286} + \left(\frac{p_3}{p_2}\right)^{0.286} + \left(\frac{p_4}{p_3}\right)^{0.286}$$

$p_1 = 1 \text{ atm}$ and $p_4 = 10 \text{ atm}$. The minimum occurs at a pressure ratio for each stage of $\sqrt[3]{10}$. Is f convex for $1 \leq p_2 \leq 10, 1 \leq p_3 \leq 10$?

4.32 In the following problem

(a) Is the objective function convex? (b) Is the constraint region convex?

$$\begin{aligned} \text{Minimize:} \quad & f(\mathbf{x}) = 100x_1 + \frac{200}{x_1x_2} \\ \text{Subject to:} \quad & g(\mathbf{x}) = 2x_2 + \frac{300}{x_1 + x_2} \geq 1 \\ & x_1 \geq 0 \\ & x_2 \geq 0 \end{aligned}$$

4.33 Answer the questions below for the following problem; in each case *justify* your answer.

$$\begin{aligned} \text{Minimize:} \quad & f(\mathbf{x}) = \frac{1}{4}x_1^4 - \frac{1}{2}x_1^2 - x_2 \\ \text{Subject to:} \quad & x_1^2 + x_2^2 = 4 \\ & x_1 - x_2 \leq 2 \end{aligned}$$

(a) Is the problem a convex programming problem?

(b) Is the point $\mathbf{x} = [1 \ 1]^T$ a feasible point?

(c) Is the point $\mathbf{x} = [2 \ 2]^T$ an interior point?

4.34 Happel and Jordan (1975) reported an objective function (cost) for the design of a distillation column as follows:

$$\begin{aligned} f = & 14720(100 - P) + 6560R - 30.2PR + 6560 - 30.2P \\ & + 19.5n(5000R - 23PR + 5000 - 23P)^{0.5} \\ & + 23.2[5000R - 23PR + 5000 - 23P]^{0.62} \end{aligned}$$

where n = number of theoretical stages

R = reflux ratio

P = percent recovery in bottoms stream

They reported the optimum occurs at $R = 8$, $n = 55$, and $P = 99$. Is f convex at this point? Are there nearby regions where f is not convex?

4.35 Given a linear objective function,

$$f = x_1 + x_2$$

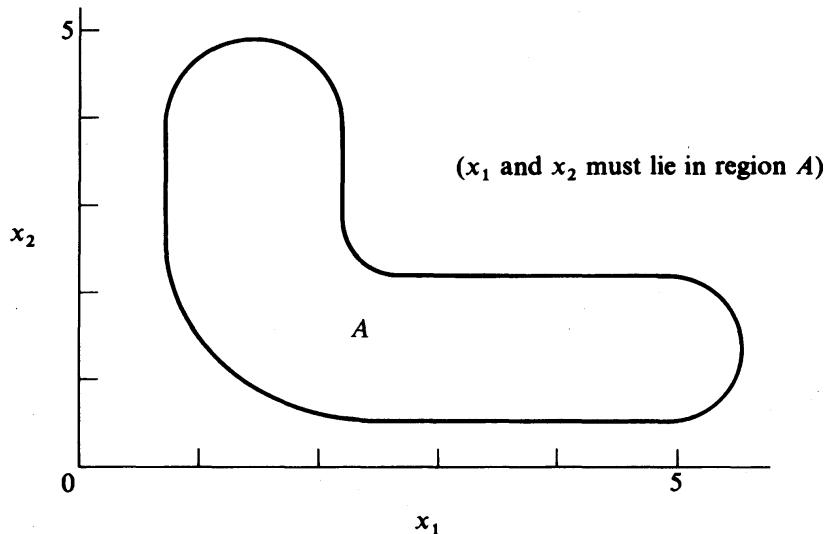


FIGURE P4.35

explain why a nonconvex region such as region A in Figure P4.35 causes difficulties in the search for the maximum of f in the region. Why is region A not convex?

4.36 Consider the following objective function

$$f(\alpha) = \sum_{i=1}^n |x_i - \alpha|$$

Show that f is convex. *Hint:* Expand f for both n odd and n even. You can plot the function to assist in your analysis. Under what circumstances is

$$f(x) = \sum_{i=1}^n c_i |x - \alpha_i|$$

convex?

4.37 Classify the stationary points of

(a) $f = -x^4 + x^3 + 20$

(b) $f = x^3 + 3x^2 + x + 5$

(c) $f = x^4 - 2x^2 + 1$

(d) $f = x_1^2 - 8x_1x_2 + x_2^2$

according to Table 4.2

4.38 List stationary points and their classification (maximum, minimum, saddle point) of

(a) $f = x_1^2 + 2x_1 + 3x_2^2 + 6x_2 + 4$

(b) $f = x_1 + x_2 + x_1^2 - 4x_1x_2 + 2x_2^2$

4.39 State what type of surface is represented by

$$f(\mathbf{x}) = 2x_1^2x_2 - 2x_2^2 + x_1^3$$

at the stationary point $\mathbf{x} = [0 \ 0]^T$ (use Table 4.2).

4.40 Interpret the geometry of the following function at its stationary point in terms of Table 4.2

$$f(\mathbf{x}) = 3x_1^3x_2$$

4.41 Classify the following function in terms of the list in Table 4.2:

$$f(\mathbf{x}) = 10x_1 - x_1^2 + 10x_2 - x_2^2 - x_1x_2 + x_1^3 - 34$$

4.42 In crystal NaCl, each Na^+ or Cl^- ion is surrounded by 6 nearest neighbors of opposite charge and 12 nearest neighbors of the same charge. Two sets of forces oppose each other: the coulombic attraction and the hard-core repulsion. The potential energy $u(r)$ of the crystal is given by the Lennard–Jones potential expression,

$$u(r) = 4\epsilon \left[\left(\frac{\sigma}{r} \right)^{12} - \left(\frac{\sigma}{r} \right)^6 \right]$$

where ϵ, σ are constants, such that $\epsilon > 0, \sigma > 0$.

(a) Does the Lennard–Jones potential $u(r)$ have a stationary point(s)? If it does, locate it (them).

(b) Identify the nature of the stationary point(s) min, max, etc.

(c) What is the magnitude of the potential energy at the stationary points?

4.43 Consider the function

$$y = (x - a)^2$$

Note that $x = a$ minimizes y . Let $z = x^2 - 4x + 16$. Does the solution to $x^2 - 4x + 16 = 0$,

$$x = \frac{4 \pm \sqrt{-48}}{2} = 2 \pm j2\sqrt{3}$$

minimize z ? ($j = \sqrt{-1}$).

4.44 The following objective function can be seen by inspection to have a minimum at $x = 0$:

$$f(x) = |x^3|$$

Can the criteria of Section 4.5 be applied to test this outcome?

4.45 (a) Consider the objective function,

$$f = 6x_1^2 + x_2^3 + 6x_1x_2 + 3x_2^2$$

Find the stationary points and classify them using the Hessian matrix.

(b) Repeat for

$$f = 3x_1^2 + 6x_1 + x_2^2 + 6x_1x_2 + x_3 + 2x_3^2 + x_2x_3 + x_2$$

(c) Repeat for

$$f = a_0x_1 + a_1x_2 + a_2x_1^2 + a_3x_2^2 + a_4x_1x_2$$

4.46 An objective function is

$$f(\mathbf{x}) = (x_1 - 8)^2 + (x_2 - 5)^2 + 16$$

By inspection, you can find $\mathbf{x}^* = [8 \ 5]^T$ yields the minimum of $f(\mathbf{x})$. Show that \mathbf{x}^* meets the necessary and sufficient conditions for a minimum.

4.47 Analyze the function

$$f(\mathbf{x}) = \frac{1}{4}x^4 - \frac{1}{2}x^2$$

Find all of its stationary points and determine if they are maxima, minima, or inflection (saddle) points. Sketch the curve in the region of

$$-2 \leq x \leq 2$$

4.48 Determine if the following objective function

$$f(\mathbf{x}) = 2x_1^3 + x_2^2 + x_1^2x_2^2 + 4x_1x_2 + 3$$

has local minima or maxima. Classify each point clearly.

4.49 Is the following function unimodal (only one extremum) or multimodal (more than one extremum)?

$$f(\mathbf{x}) = \begin{cases} -x^2 & -\infty \leq x \leq 0 \\ -x^2 & 0 \leq x \leq 1 \\ e^{x-1} & 1 \leq x \leq \infty \end{cases}$$

4.50 Determine whether the solution $\mathbf{x} = [-0.87 \ -0.8]^T$ for the objective function

$$f(x) = x_1^4 + 12x_2^3 - 15x_1^2 - 56x_2 + 60$$

is indeed a maximum.

**OPTIMIZATION OF UNCONSTRAINED
FUNCTIONS: ONE-DIMENSIONAL SEARCH**

5.1 Numerical Methods for Optimizing a Function of One Variable 155
5.2 Scanning and Bracketing Procedures 156
5.3 Newton and Quasi-Newton Methods of Unidimensional Search 157
5.4 Polynomial Approximation Methods 166
5.5 How One-Dimensional Search Is Applied in a Multidimensional Problem ... 173
5.6 Evaluation of Unidimensional Search Methods 176
References 176
Supplementary References 177
Problems 177

A GOOD TECHNIQUE for the optimization of a function of just one variable is essential for two reasons:

1. Some unconstrained problems inherently involve only one variable
2. Techniques for unconstrained and constrained optimization problems generally involve repeated use of a one-dimensional search as described in Chapters 6 and 8.

Prior to the advent of high-speed computers, methods of optimization were limited primarily to analytical methods, that is, methods of calculating a potential extremum were based on using the necessary conditions and analytical derivatives as well as values of the objective function. Modern computers have made possible iterative, or numerical, methods that search for an extremum by using function and sometimes derivative values of $f(x)$ at a sequence of trial points x^1, x^2, \dots .

As an example consider the following function of a single variable x (see Figure 5.1).

$$f(x) = x^2 - 2x + 1$$

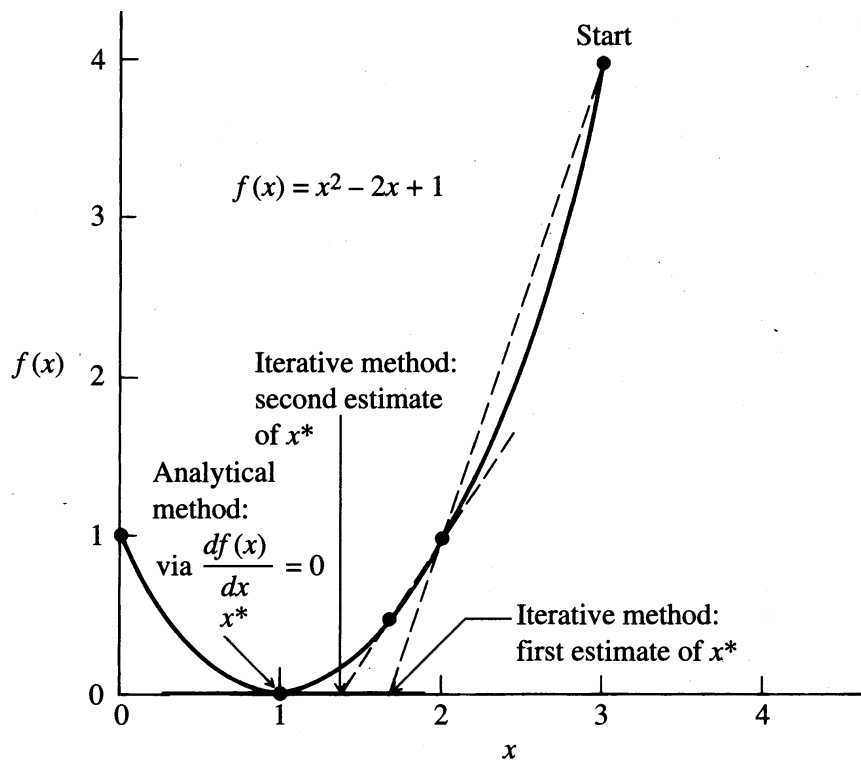


FIGURE 5.1

Iterative versus analytical methods of finding a minimum.

An analytical method of finding x^* at the minimum of $f(x)$ is to set the gradient of $f(x)$ equal to zero

$$\frac{df(x)}{dx} = 0 = 2x - 2$$

and solve the resulting equation to get $x^* = 1$; x^* can be tested for the sufficient conditions to ascertain that it is indeed a minimum:

$$\frac{d^2f(1)}{dx^2} = 2 > 0$$

To carry out an iterative method of numerical minimization, start with some initial value of x , say $x^0 = 0$, and calculate successive values of $f(x) = x^2 - 2x + 1$ and possibly df/dx for other values of x , values selected according to whatever strategy is to be employed. A number of different strategies are discussed in subsequent sections of this chapter. Stop when $f(x^{k+1}) - f(x^k) < \varepsilon_1$ or when

$$\left. \frac{df}{dx} \right|_{x^k} < \varepsilon_2$$

where the superscript k designates the iteration number and ε_1 and ε_2 are the pre-specified tolerances or criteria of precision.

If $f(x)$ has a simple closed-form expression, analytical methods yield an exact solution, a closed form expression for the optimal x , x^* . If $f(x)$ is more complex, for example, if it requires several steps to compute, then a numerical approach must be used. Software for nonlinear optimization is now so widely available that the numerical approach is almost always used. For example, the “Solver” in the Microsoft Excel spreadsheet solves linear and nonlinear optimization problems, and many FORTRAN and C optimizers are available as well. General optimization software is discussed in Section 8.9.

Analytical methods are usually difficult to apply for nonlinear objective functions with more than one variable. For example, suppose that the nonlinear function $f(\mathbf{x}) = f(x_1, x_2, \dots, x_n)$ is to be minimized. The necessary conditions to be used are

$$\frac{\partial f(\mathbf{x})}{\partial x_1} = 0$$

$$\frac{\partial f(\mathbf{x})}{\partial x_2} = 0$$

⋮

$$\frac{\partial f(\mathbf{x})}{\partial x_n} = 0$$

Each of the partial derivatives when equated to zero may well yield a nonlinear equation. Hence, the minimization of $f(\mathbf{x})$ is converted into a problem of solving a set of nonlinear equations in n variables, a problem that can be just as difficult to solve as the original problem. Thus, most engineers prefer to attack the minimization problem directly by one of the numerical methods described in Chapter 6, rather than to use an indirect method. Even when minimizing a function of one variable by an indirect method, using the necessary conditions can lead to having to find the real roots of a nonlinear equation.

5.1 NUMERICAL METHODS FOR OPTIMIZING A FUNCTION OF ONE VARIABLE

Most algorithms for unconstrained and constrained optimization make use of an efficient unidimensional optimization technique to locate a local minimum of a function of one variable. Nash and Soter (1996) and other general optimization books (e.g., Dennis and Schnabel, 1983) have reviewed one-dimensional search techniques that calculate the interval in which the minimum of a function lies. To apply these methods you initially need to know an initial bracket Δ^0 that contains the minimum of the objective function $f(x)$, and that $f(x)$ is unimodal in the interval. This can be done by coding the function in a spreadsheet or in a programming language like Visual Basic, Fortran, or C, choosing an interval, and evaluating $f(x)$ at a grid of points in that interval. The interval is extended if the minimum is at an end point. There are various methods of varying the initial interval to reach a final interval Δ^n . In the next section we describe a few of the methods that prove to be the most effective in practice.

One method of optimization for a function of a single variable is to set up as fine a grid as you wish for the values of x and calculate the function value for every point on the grid. An approximation to the optimum is the best value of $f(x)$. Although this is not a very efficient method for finding the optimum, it can yield acceptable results. On the other hand, if we were to utilize this approach in optimizing a multivariable function of more than, say, five variables, the computer time is quite likely to become prohibitive, and the accuracy is usually not satisfactory.

In selecting a search method to minimize or maximize a function of a single variable, the most important concerns are software availability, ease of use, and efficiency. Sometimes the function may take a long time to compute, and then efficiency becomes more important. For example, in some problems a simulation may be required to generate the function values, such as in determining the optimal number of trays in a distillation column. In other cases you have no functional description of the physical-chemical model of the process to be optimized and are forced to operate the process at various input levels to evaluate the value of the process output. The generation of a new value of the objective function in such circumstances may be extremely costly, and no doubt the number of plant tests would be limited and have to be quite judiciously designed. In such circumstances, efficiency is a key criterion in selecting a minimization strategy.

5.2 SCANNING AND BRACKETING PROCEDURES

Some unidimensional search procedures require that a bracket of the minimum be obtained as the first part of the strategy, and then the bracket is narrowed. Along with the statement of the objective function $f(x)$ there must be some statement of bounds on x or else the implicit assumption that x is unbounded ($-\infty < x < \infty$). For example, the problem

$$\text{Minimize: } f(x) = (x - 100)^2$$

has an optimal value of $x^* = 100$. Clearly you would not want to start at $-\infty$ (i.e., a large negative number) and try to bracket the minimum. Common sense suggests estimating the minimum x and setting up a sufficiently wide bracket to contain the true minimum. Clearly, if you make a mistake and set up a bracket of $0 \leq x \leq 10$, you will find that the minimum occurs at one of the bounds, hence the bracket must be revised. In engineering and scientific work physical limits on temperature, pressure, concentration, and other physically meaningful variables place practical bounds on the region of search that might be used as an initial bracket.

Several strategies exist for scanning the independent variable space and determining an acceptable range for search for the minimum of $f(x)$. As an example, in the above function, if we discretize the independent variable by a grid spacing of 0.01, and then initiate the search at zero, proceeding with consecutively higher values of x , much time and effort would be consumed in order to set up the initial bracket for x . Therefore, acceleration procedures are used to scan rapidly for a suitable range of x . One technique might involve using a functional transformation (e.g., $\log x$) in order to look at wide ranges of the independent variable. Another method might be to use a variable grid spacing. Consider a sequence in x given by the following formula:

$$x^{k+1} = x^k + \delta \cdot 2^{k-1}. \quad (5.1)$$

Equation (5.1) allows for successively wider-spaced values, given some base increment (delta). Table 5.1 lists the values of x and $f(x) = (x - 100)^2$ for Equation (5.1) with $\delta = 1$. Note that in nine calculations we have bounded the minimum of $f(x)$. Another scanning procedure could be initiated between $x = 63$ and $x = 255$, with δ reduced, and so on to find the minimum of $f(x)$. However, more efficient techniques are discussed in subsequent sections of this chapter.

In optimization of a function of a single variable, we recognize (as for general multivariable problems) that there is no substitute for a good first guess for the starting point in the search. Insight into the problem as well as previous experience

TABLE 5.1
Acceleration in fixing an initial bracket

x	0	1	3	7	15	31	63	127	255
$f(x)$	10^4	9801	9409	8649	7225	4761	1369	729	2325

are therefore often very important factors influencing the amount of time and effort required to solve a given optimization problem.

The methods considered in the rest of this chapter are generally termed descent methods for minimization because a given step is pursued only if it yields an improved value for the objective function. First we cover methods that use function values or first or second derivatives in Section 5.3, followed by a review of several methods that use only function values in Section 5.4.

5.3 NEWTON AND QUASI-NEWTON METHODS OF UNIDIMENSIONAL SEARCH

Three basic procedures for finding an extremum of a function of one variable have evolved from applying the necessary optimality conditions to the function:

1. Newton's method
2. Finite difference approximation of Newton's method
3. Quasi-Newton methods

In comparing the effectiveness of these techniques, it is useful to examine the rate of convergence for each method. Rates of convergence can be expressed in various ways, but a common classification is as follows:^a

Linear

$$\frac{\|\mathbf{x}^{k+1} - \mathbf{x}^*\|}{\|\mathbf{x}^k - \mathbf{x}^*\|} \leq c \quad 0 \leq c < 1, k \text{ large} \quad (5.2)$$

(rate usually slow in practice)

Order p

$$\frac{\|\mathbf{x}^{k+1} - \mathbf{x}^*\|}{\|\mathbf{x}^k - \mathbf{x}^*\|^p} \leq c \quad c \geq 0, p \geq 1, k \text{ large} \quad (5.3)$$

(rate fastest in practice if $p > 1$)

If $p = 2$, the order of convergence is said to be quadratic.

To understand these definitions, assume that the algorithm generating the sequence of points \mathbf{x}^k is converging to \mathbf{x}^* , that is, as $k \rightarrow \infty$, if Equation (5.2) holds for large k , $\mathbf{x}^k \rightarrow \mathbf{x}^*$. Then

$$\|\mathbf{x}^{k+1} - \mathbf{x}^*\| \leq c \|\mathbf{x}^k - \mathbf{x}^*\| \quad k \text{ large}$$

^aThe symbols \mathbf{x}^k , \mathbf{x}^{k+1} , and so on refer to the k th or $(k + 1)$ st stage of iteration and not to powers of \mathbf{x} .

so the error at iteration $k + 1$ is bounded by c times the error at iteration k , where $c < 1$. If $c = 0.1$, then the error is reduced by a factor of 10 at each iteration, at least for the later iterations. The constant c is called the convergence ratio.

If Equation (5.3) holds for large k , then $\|\mathbf{x}^{k+1} - \mathbf{x}^*\| \leq c \|\mathbf{x}^k - \mathbf{x}^*\|^p$, k large enough. If $p = 2$, and $\|\mathbf{x}^k - \mathbf{x}^0\| = 10^{-1}$ for some k , then

$$\|\mathbf{x}^{k+1} - \mathbf{x}^*\| \leq c \cdot 10^{-2}$$

$$\|\mathbf{x}^{k+2} - \mathbf{x}^*\| \leq c^2 \cdot 10^{-4}$$

$$\|\mathbf{x}^{k+3} - \mathbf{x}^*\| \leq c^3 \cdot 10^{-6}$$

and so on.

Hence, if c is around 1.0, the error decreases very rapidly, the number of correct digits in \mathbf{x}^k doubling with each iteration. Because all real numbers in double precision arithmetic have about 16 significant decimal digits, only a few iterations are needed before the limits of accuracy of Equation (5.3) are reached.

Superlinear

$$\lim_{k \rightarrow \infty} \frac{\|\mathbf{x}^{k+1} - \mathbf{x}^*\|}{\|\mathbf{x}^k - \mathbf{x}^*\|} \rightarrow 0 \quad (\text{or } < c_k \text{ and } c_k \rightarrow 0 \text{ as } k \rightarrow \infty) \quad (5.4)$$

(rate usually fast in practice)

For a function of a single variable $\|\mathbf{x}\| = |x|$ itself.

5.3.1 Newton's Method

Recall that the first-order necessary condition for a local minimum is $f'(x) = 0$. Consequently, you can solve the equation $f'(x) = 0$ by Newton's method to get

$$x^{k+1} = x^k - \frac{f'(x^k)}{f''(x^k)} \quad (5.5)$$

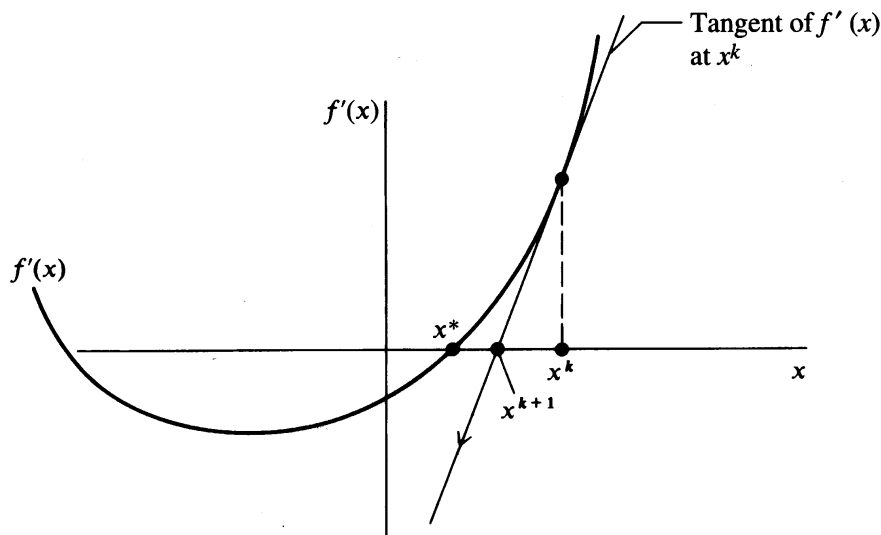
making sure on each stage k that $f(x^{k+1}) < f(x^k)$ for a minimum. Examine Figure 5.2.

To see what Newton's method implies about $f(x)$, suppose $f(x)$ is approximated by a quadratic function at x^k

$$f(x) = f(x^k) + f'(x^k)(x - x^k) + \frac{1}{2}f''(x^k)(x - x^k)^2 \quad (5.6)$$

Find $df(x)/dx = 0$, a stationary point of the quadratic model of the function. The result obtained by differentiating Equation (5.6) with respect to x is

$$f'(x^k) + \left(\frac{1}{2}\right)(2)f''(x^k)(x - x^k) = 0 \quad (5.7)$$

**FIGURE 5.2**

Newton's method applied to the solution of $f'(x) = 0$.

which can be rearranged to yield Equation (5.5). Consequently, Newton's method is equivalent to using a quadratic model for a function in minimization (or maximization) and applying the necessary conditions.

The advantages of Newton's method are

1. The procedure is locally quadratically convergent [$p = 2$ in Equation (5.3)] to the extremum as long as $f''(x) \neq 0$.
2. For a quadratic function, the minimum is obtained in one iteration.

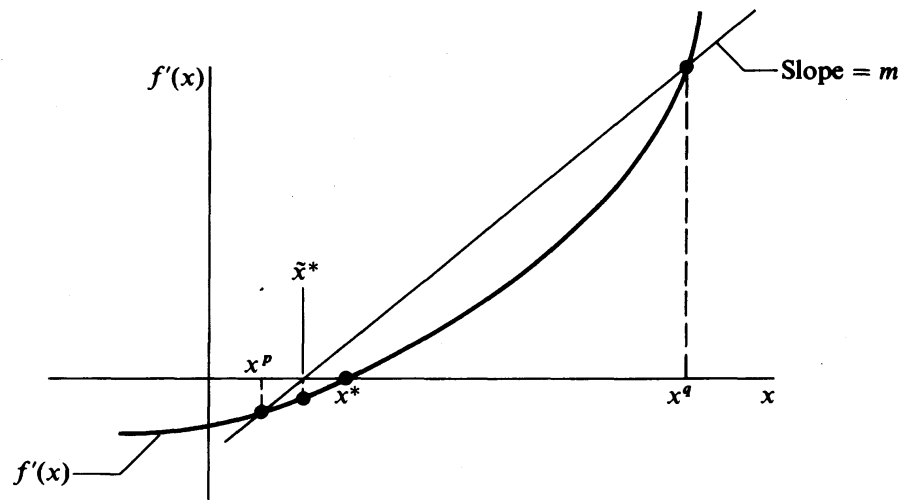
The disadvantages of the method are

1. You have to calculate both $f'(x)$ and $f''(x)$.
2. If $f''(x) \rightarrow 0$, the method converges slowly.
3. If the initial point is not close enough to the minimum, the method as described earlier will not converge. Modified versions that guarantee convergence from poor starting points are described in Bazarra et al. (1993) and Nash and Sofer (1996).

5.3.2 Finite Difference Approximations to Derivatives

If $f(x)$ is not given by a formula, or the formula is so complicated that analytical derivatives cannot be formulated, you can replace Equation (5.5) with a finite difference approximation

$$x^{k+1} = x^k - \frac{[f(x+h) - f(x-h)]/2h}{[f(x+h) - 2f(x) + f(x-h)]/h^2} \quad (5.8)$$

**FIGURE 5.3**

Quasi-Newton method for solution of $f'(x) = 0$.

Central differences were used in Equation (5.8), but forward differences or any other difference scheme would suffice as long as the step size h is selected to match the difference formula and the computer (machine) precision with which the calculations are to be executed. The main disadvantage is the error introduced by the finite differencing.

5.3.3 Quasi-Newton Method

In the quasi-Newton method (secant method) the approximate model analogous to Equation (5.7) to be solved is

$$f'(x^k) + m(x - x^k) = 0 \quad (5.9)$$

where m is the slope of the line connecting the point x^p and a second point x^q , given by

$$m = \frac{f'(x^q) - f'(x^p)}{x^q - x^p}$$

The quasi-Newton approximates $f'(x)$ as a straight line (examine Figure 5.3); as $x^q \rightarrow x^p$, m approaches the second derivative of $f(x)$. Thus Equation (5.9) imitates Newton's method

$$\tilde{x} = x^q - \frac{f'(x^q)}{[f'(x^q) - f'(x^p)]/(x^q - x^p)} \quad (5.10)$$

where \tilde{x} is the approximation to x^* achieved on one iteration k . Note that $f'(x)$ can itself be approximated by finite differencing.

Quasi-Newton methods start out by using two points x^p and x^q spanning the interval of x , points at which the first derivatives of $f(x)$ are of opposite sign. The zero of Equation (5.9) is predicted by Equation (5.10), and the derivative of the function is then evaluated at the new point. The two points retained for the next step are \tilde{x} and either x^q or x^p . This choice is made so that the pair of derivatives $f'(\tilde{x})$, and either $f'(x^p)$ or $f'(x^q)$, have opposite signs to maintain the bracket on x^* . This variation is called “regula falsi” or the method of false position. In Figure 5.3, for the $(k + 1)$ st search, \tilde{x} and x^q would be selected as the end points of the secant line.

Quasi-Newton methods may seem crude, but they work well in practice. The order of convergence is $(1 + \sqrt{5})/2 \approx 1.6$ for a single variable. Their convergence is slightly slower than a properly chosen finite difference Newton method, but they are usually more efficient in terms of total function evaluations to achieve a specified accuracy (see Dennis and Schnabel, 1983, Chapter 2).

For any of the three procedures outlined in this section, in minimization you assume the function is unimodal, bracket the minimum, pick a starting point, apply the iteration formula to get x^{k+1} (or \tilde{x}) from x^k (or x^p and x^q), and make sure that $f(x^{k+1}) < f(x^k)$ on each iteration so that progress is made toward the minimum. As long as $f''(x^k)$ or its approximation is positive, $f(x)$ decreases.

Of course, you must start in the correct direction to reduce $f(x)$ (for a minimum) by testing an initial perturbation in x . For maximization, minimize $-f(x)$.

EXAMPLE 5.1 COMPARISON OF NEWTON, FINITE DIFFERENCE NEWTON, AND QUASI-NEWTON METHODS APPLIED TO A QUADRATIC FUNCTION

In this example, we minimize a simple quadratic function $f(x) = x^2 - x$ that is illustrated in Figure E5.1a using one iteration of each of the methods presented in Section 5.3.

Solution. By inspection we can pick a bracket on the minimum, say $x = -3$ to $x = 3$. Assume $x^0 = 3$ is the starting point for the minimization.

Newton’s method. For Newton’s method sequentially apply Equation (5.5). Examine Figure 5.1b for $f(x) = x^2 - x$ and $f'(x) = 2x - 1$; $f''(x) = 2$. Note $f''(x)$ is always positive-definite. For this example Equation (5.5) is

$$x^1 = x^0 - \frac{f'(x^0)}{f''(x^0)} \quad (a)$$

and

$$x^1 = 3 - \frac{5}{2} = 0.5$$

Because the function is quadratic and hence $f'(x)$ is linear, the minimum is obtained in one step. If the function were not quadratic, then additional iterations using Equation (5.5) would take place.

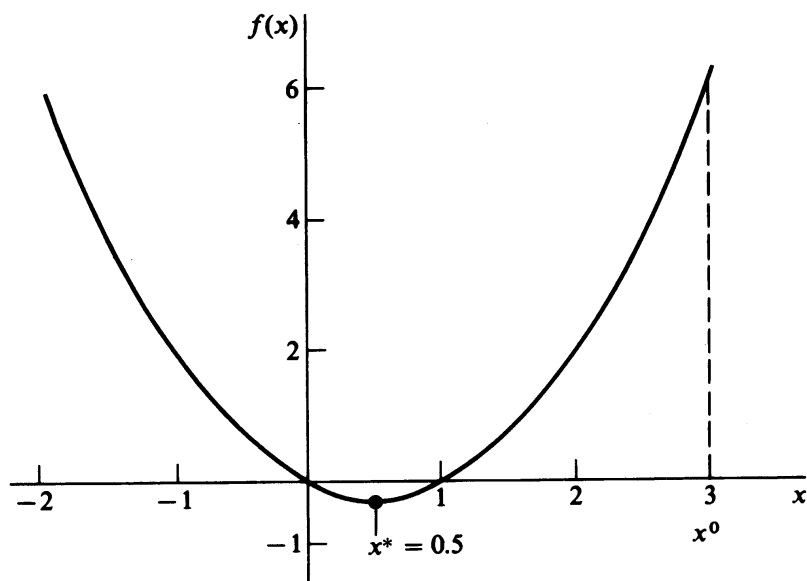


FIGURE E5.1a

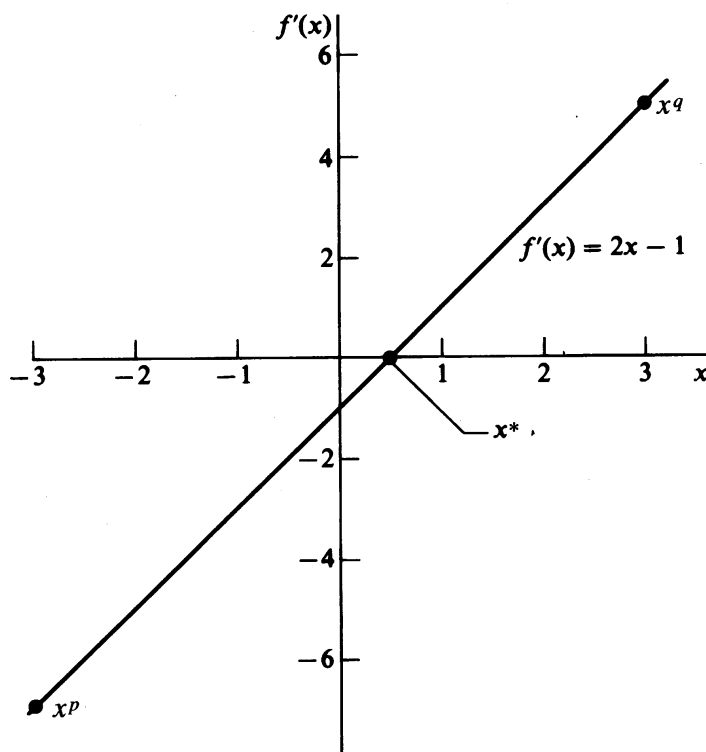


FIGURE E5.1b

Finite difference Newton method. Application of Equation (5.8) to $f(x) = x^2 - x$ is illustrated here. However, we use a forward difference formula for $f'(x)$ and a three-point central difference formula for $f''(x)$

$$x^{k+1} = x^k - \frac{[f(x+h) - f(x)]/h}{[f(x+h) - 2f(x) + f(x-h)]/h^2} \quad (b)$$

with $h = 10^{-3}$:

$$\begin{aligned} x^1 &= 3 - \frac{[f(3.001) - f(3.0)]/10^{-3}}{[f(3.001) - 2f(3.0) + f(2.999)]/(10^{-3})^2} \\ &= 3 - (10^{-3}) \frac{(6.005001 - 6.000000)}{(6.005001 - 12.000000 + 5.995001)} \\ &= 3 - (10^{-3}) \frac{0.005001}{0.000002} = 3 - 2.500500 \\ &= 0.499500 \end{aligned}$$

One more iteration could be taken to improve the estimate of x^* , perhaps with a smaller value of h (if desired).

Quasi-Newton method. The application of Equation (5.10) to $f(x) = x^2 - x$ starts with the two points $x = -3$ and $x = 3$ corresponding to the x^p and x^q , respectively, in Figure 5.3:

$$f'(-3) = -7 \quad f'(3) = 5$$

$$x^1 = 3 - \frac{5}{[5 - (-7)]/[3 - (-3)]} = 3 - 2.5 = 0.5$$

As before, the optimum is reached in one step because $f'(x)$ is linear, and the linear extrapolation is valid.

EXAMPLE 5.2 MINIMIZING A MORE DIFFICULT FUNCTION

In this example we minimize a nonquadratic function $f(x) = x^4 - x + 1$ that is illustrated in Figure E5.2a, using the same three methods as in Example 5.1. For a starting point of $x = 3$, minimize $f(x)$ until the change in x is less than 10^{-7} . Use $h = 0.1$ for the finite-difference method. For the quasi-Newton method, use $x^q = 3$ and $x^p = -3$.

Solution

Newton's method. For Newton's method, $f' = 4x^3 - 1$ and $f'' = 12x^2$, and the sequence of steps is

$$x_1 = x_0 - \frac{4x_0^3 - 1}{12x_0^2} \tag{a}$$

$$= 3 - \frac{107}{108} = 2.009259$$

$$x_2 = 2.00926 - \frac{31.4465}{48.4454} = 1.36015$$

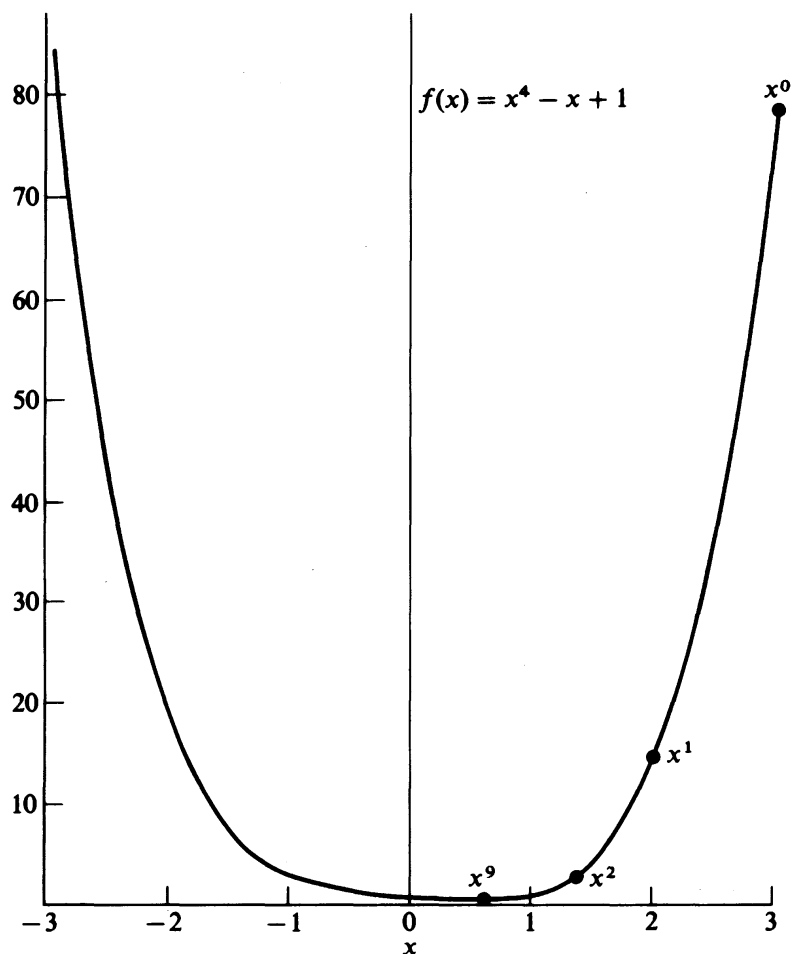


FIGURE E5.2a
Newton iterates for fourth order function.

Additional iterations yield the following values for x :

k	x^k	$\frac{x^{k+1} - x^*}{x^k - x^*}$	$\frac{x^{k+1} - x^*}{ x^k - x^* ^2}$
0	3.00000		
1	2.009259	0.582	0.246
2	1.3601480	0.529	0.384
3	0.9518103	0.441	0.604
4	0.7265254	0.300	0.932
5	0.6422266	0.127	1.315
6	0.6301933	0.019	1.547
7	0.6299606	0.000	1.587
8	0.6299605		
9	0.6299605		

As you can see from the third and fourth columns in the table the rate of convergence of Newton's method is superlinear (and in fact quadratic) for this function.

Finite Difference Newton. Equation (5.8) for this example is

$$x^{k+1} = x^k - \frac{h}{2} \frac{[f(x+h) - f(x-h)]}{[f(x+h) - 2f(x) + f(x-h)]} \tag{b}$$

For the same problem as used in Newton's method, the first iteration using (b) for $h = 10^{-4}$ is

$$x^1 = 3 - \left[\frac{10^{-4}}{2} \right] \frac{[f(3.0001) - f(2.9999)]}{[f(3.0001) - 2f(3.000) + f(2.999)]}$$

Other values of h give

k	x^k		
	$h = 0.10$	$h = 10^{-4}$	$h = 10^{-7}$
0	3.00000	2.00926	3.00000
1	2.00833	1.36015	2.21568
2	1.35816	0.951811	1.46785
3	0.948531	0.726526	0.955459
4	0.721882	0.642227	0.736528
5	0.636823	0.630193	0.642986
6	0.624849	0.629960	0.631846
7	0.624668	0.6299605191	0.630035
8	0.624669	0.629964
9	0.624669313	0.629961
10	0.629961
11	0.629960525

For $h = 10^{-8}$, the procedure diverged after the second iteration.

Quasi-Newton. The application of Equation (5.10) yields the following results (examine Figure E5.2b). Note how the shape of $f'(x)$ implies that a large number of iterations are needed to reach x^* . Some of the values of $f'(x)$ and x during the search are shown in the following table; notice that x^q remains unchanged in order to maintain the bracket with $f'(x) > 0$.

k	x^q	x^p	$f'(x^p)$
0	3.0	-3.0	-109.0000
1	3.0	0.0277	-0.9991
2	3.0	0.0552	-0.9992
3	3.0	0.0825	-0.9977
4	3.0	0.1094	-0.9899
5	3.0	0.1361	-0.9899
20	3.0	0.4593	-0.6124
50	3.0	0.6223	-0.0360
100	3.0	0.6299	-1.399×10^{-4}
132	3.0	0.6299	-3.952×10^{-6}

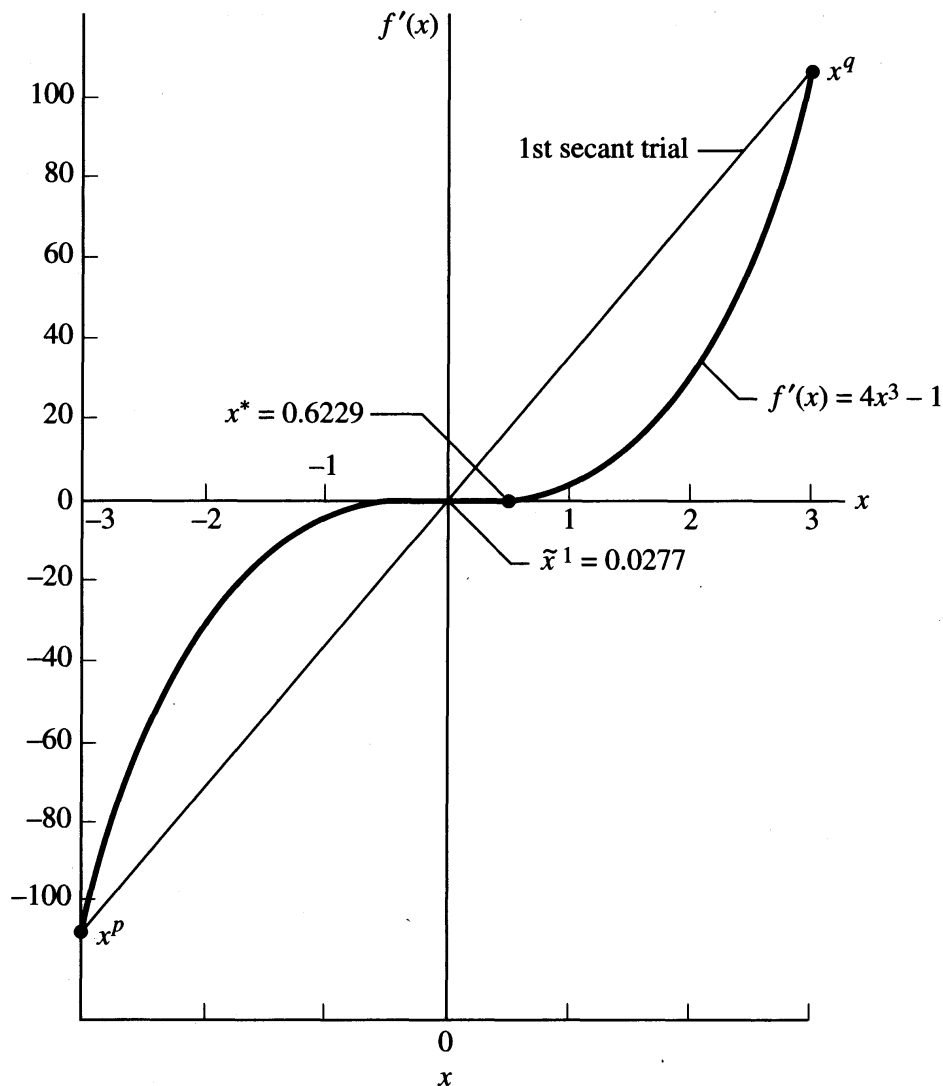


FIGURE E5.2b
Quazi-Newton method applied to $f'(x)$.

5.4 POLYNOMIAL APPROXIMATION METHODS

Another class of methods of unidimensional minimization locates a point x near x^* , the value of the independent variable corresponding to the minimum of $f(x)$, by extrapolation and interpolation using polynomial approximations as models of $f(x)$. Both quadratic and cubic approximation have been proposed using function values only and using both function and derivative values. In functions where $f'(x)$ is continuous, these methods are much more efficient than other methods and are now widely used to do line searches within multivariable optimizers.

5.4.1 Quadratic Interpolation

We start with three points x_1 , x_2 , and x_3 in increasing order that might be equally spaced, but the extreme points must bracket the minimum. From the analysis in Chapter 2, we know that a quadratic function $f(x) = a + bx + cx^2$ can be passed

exactly through the three points, and that the function can be differentiated and the derivative set equal to 0 to yield the minimum of the approximating function

$$\tilde{x} = -\frac{b}{2c} \quad (5.11)$$

Suppose that $f(x)$ is evaluated at $x_1, x_2,$ and x_3 to yield $f(x_1) \equiv f_1, f(x_2) \equiv f_2,$ and $f(x_3) \equiv f_3$. The coefficients b and c can be evaluated from the solution of the three linear equations

$$f(x_1) = a + bx_1 + cx_1^2$$

$$f(x_2) = a + bx_2 + cx_2^2$$

$$f(x_3) = a + bx_3 + cx_3^2$$

via determinants or matrix algebra. Introduction of b and c expressed in terms of $x_1, x_2, x_3, f_1, f_2,$ and f_3 into Equation (5.11) gives

$$\tilde{x}^* = \frac{1}{2} \left[\frac{(x_2^2 - x_3^2)f_1 + (x_3^2 - x_1^2)f_2 + (x_1^2 - x_2^2)f_3}{(x_2 - x_3)f_1 + (x_3 - x_1)f_2 + (x_1 - x_2)f_3} \right] \quad (5.12)$$

To illustrate the first stage in the search procedure, examine the four points in Figure 5.4 for stage 1. We want to reduce the initial interval $[x_1, x_3]$. By examining the values of $f(x)$ [with the assumptions that $f(x)$ is unimodal and has a minimum], we can discard the interval from x_1 to x_2 and use the region (x_2, x_3) as the new interval. The new interval contains three points, (x_2, \tilde{x}, x_3) that can be introduced into Equation (5.12) to estimate a x^* , and so on. In general, you evaluate $f(x^*)$ and discard from the set $\{x_1, x_2, x_3\}$ the point that corresponds to the greatest value of $f(x)$, unless

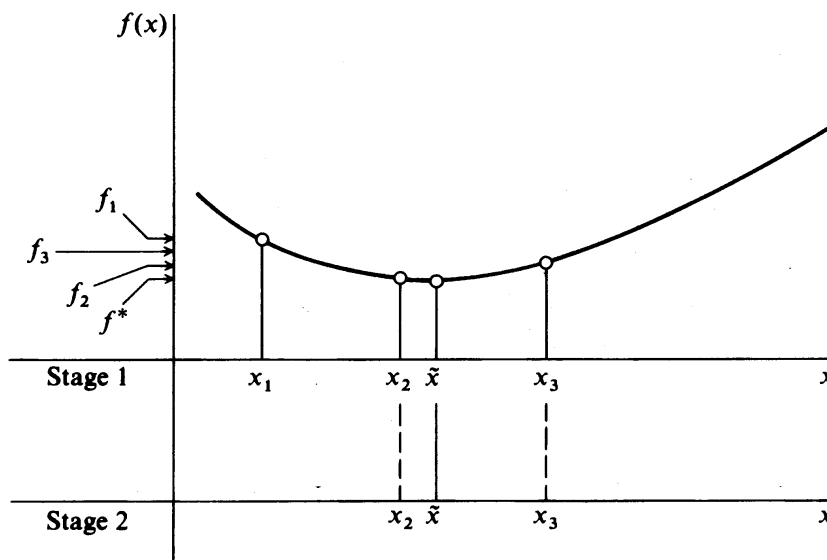
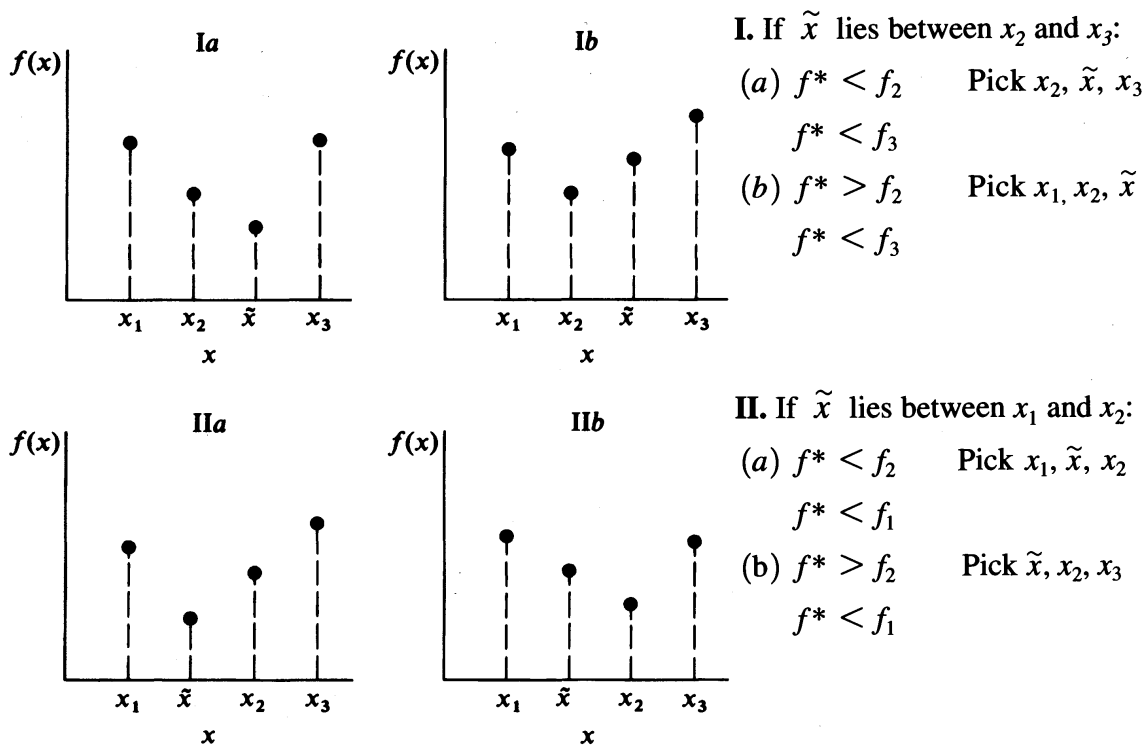


FIGURE 5.4
Two stages of quadratic interpolation.

**FIGURE 5.5**

How to maintain a bracket on the minimum in quadratic interpolation.

a bracket on the minimum of $f(x)$ is lost by so doing, in which case you discard the x_i so as to maintain the bracket. The specific tests and choices of x_i to maintain the bracket are illustrated in Figure 5.5. In Figure 5.5, $f^* \equiv f(\tilde{x})$. If x^* and whichever of $\{x_1, x_2, x_3\}$ corresponding to the smallest $f(x)$ differ by less than the prescribed accuracy in x , or the prescribed accuracy in the corresponding values of $f(x)$ is achieved, terminate the search. Note that only function evaluations are used in the search and that only one new function evaluation (for \tilde{x}) has to be carried out at each new iteration.

EXAMPLE 5.3 APPLICATION OF QUADRATIC INTERPOLATION

The function to be minimized is $f(x) = x^2 - x$ and is illustrated in Figure E5.1a. Three points bracketing the minimum $(-1.7, -0.1, 1.5)$ are used to start the search for the minimum of $f(x)$; we use equally spaced points here but that is not a requirement of the method.

Solution

$$\begin{array}{lll}
 x_1 = -1.7 & x_2 = -0.1 & x_3 = 1.5 \\
 f(x_1) = 4.59 & f(x_2) = 0.11 & f(x_3) = 0.75 \\
 \Delta x = 1.6 & &
 \end{array}$$

Two different formulas for quadratic interpolation can be compared: Equation (5.8), the finite difference method, and Equation (5.12).

$$\tilde{x}^* = x_2 - \frac{\Delta x[f(x_3) - f(x_1)]}{2[f(x_3) - 2f(x_2) + f(x_1)]} \quad (5.8)$$

$$= -0.1 - \frac{1.6(0.75 - 4.59)}{2(0.75 - 2(0.11) + 4.59)} = 0.50 \quad (a)$$

$$\tilde{x}^* = \frac{1}{2} \frac{[x_2^2 - x_3^2]f(x_1) + [x_3^2 - x_1^2]f(x_2) + [x_1^2 - x_2^2]f(x_3)}{(x_2 - x_3)f(x_1) + (x_3 - x_1)f(x_2) + (x_1 - x_2)f(x_3)} \quad (5.12)$$

$$= \frac{1}{2} \frac{[(-0.1)^2 - (1.5)^2](4.59) + [(1.5)^2 - (-1.7)^2](0.11) + [(-1.7)^2 - (-0.1)^2](0.75)}{[(-0.1) - (1.5)](4.59) + [(1.5) - (-1.7)](0.11) + [(-1.7) - (-0.1)](0.75)} \quad (b)$$

$$= 0.50$$

Note that a solution on the first iteration seems to be remarkable, but keep in mind that the function is quadratic so that quadratic interpolation should be good even if approximate formulas are used for derivatives.

5.4.2 Cubic Interpolation

Cubic interpolation to find the minimum of $f(x)$ is based on approximating the objective function by a third-degree polynomial within the interval of interest and then determining the associated stationary point of the polynomial

$$f(x) = a_1x^3 + a_2x^2 + a_3x + a_4$$

Four points must be computed (that bracket the minimum) to estimate the minimum, either four values of $f(x)$, or the values of $f(x)$ and the derivative of $f(x)$, each at two points.

In the former case four linear equations are obtained with the four unknowns being the desired coefficients. Let the matrix \mathbf{X} be

$$\mathbf{X} = \begin{bmatrix} x_1^3 & x_1^2 & x_1 & 1 \\ x_2^3 & x_2^2 & x_2 & 1 \\ x_3^3 & x_3^2 & x_3 & 1 \\ x_4^3 & x_4^2 & x_4 & 1 \end{bmatrix}$$

$$\mathbf{F}^T = [f(x_1) \quad f(x_2) \quad f(x_3) \quad f(x_4)]$$

$$\mathbf{A}^T = [a_1 \quad a_2 \quad a_3 \quad a_4]$$

$$\mathbf{F} = \mathbf{XA} \quad (5.13)$$

Then the extremum of $f(x)$ is obtained by setting the derivative of $f(x)$ equal to zero and solving for \tilde{x}

$$\frac{df(x)}{dx} = 3a_1x^2 + 2a_2x + a_3 = 0$$

so that

$$\tilde{x} = \frac{-2a_2 \pm \sqrt{4a_2^2 - 12a_1a_3}}{6a_1} \quad (5.14)$$

The sign to use before the square root is governed by the sign of the second derivative of $f(\tilde{x})$, that is, whether a minimum or maximum is sought. The vector \mathbf{A} can be computed from $\mathbf{XA} = \mathbf{F}$ or

$$\mathbf{A} = \mathbf{X}^{-1}\mathbf{F} \quad (5.15)$$

After the optimum point \tilde{x} is predicted, it is used as a new point in the next iteration and the point with the highest [lowest value of $f(x)$ for maximization] value of $f(x)$ is discarded.

If the first derivatives of $f(x)$ are available, only two points are needed, and the cubic function can be fitted to the two pairs of the slope and function values. These four pieces of information can be uniquely related to the four coefficients in the cubic equation, which can be optimized for predicting the new, nearly optimal data point. If (x_1, f_1, f'_1) and (x_2, f_2, f'_2) are available, then the optimum \tilde{x} is

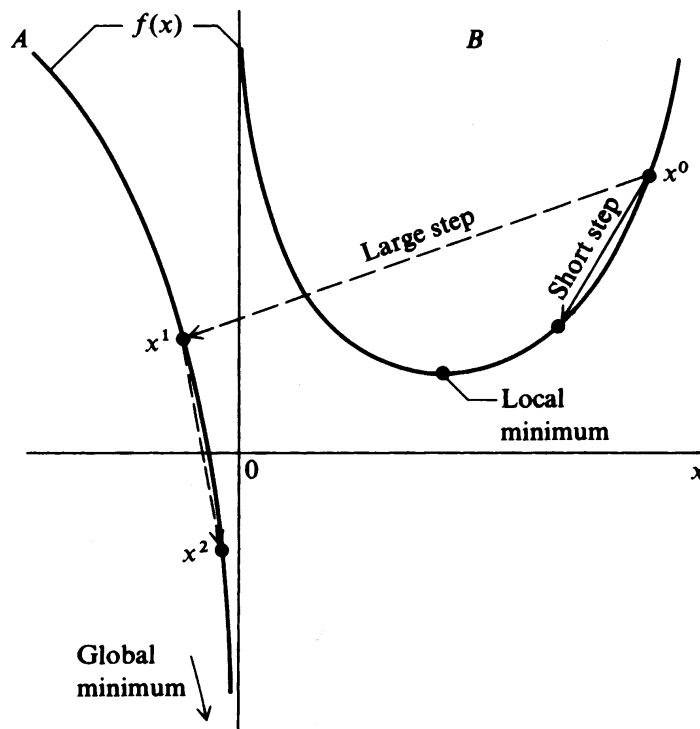
$$\tilde{x} = x_2 - \left[\frac{f'_2 + w - z}{f'_2 - f'_1 + 2w} \right] (x_2 - x_1) \quad (5.18)$$

where
$$z = \frac{3[f_1 - f_2]}{[x_2 - x_1]} + f'_1 + f'_2$$

$$w = [z^2 - f'_1 \cdot f'_2]^{1/2}$$

In a minimization problem, you require $x_1 < x_2, f'_1 < 0$, and $f'_2 > 0$ (x_1 and x_2 bracket the minimum). For the new point (\tilde{x}), calculate $f'(\tilde{x})$ to determine which of the previous two points to replace. The application of this method in nonlinear programming algorithms that use gradient information is straightforward and effective.

If the function being minimized is not unimodal locally, as has been assumed to be true in the preceding discussion, extra logic must be added to the unidimensional search code to ensure that the step size is adjusted to the neighborhood of the local optimum actually sought. For example, Figure 5.6 illustrates how a large initial step can lead to an unbounded solution to a problem when, in fact, a local minimum is sought.

**FIGURE 5.6**

A unidimensional search for a local minimum of a multimodal objective function leads to an unbounded solution.

EXAMPLE 5.4 OPTIMIZATION OF A MICROELECTRONICS PRODUCTION LINE FOR LITHOGRAPHY

You are to optimize the thickness of resist used in a production lithographic process. There are a number of competing effects in lithography.

1. As the thickness t (measured in micrometers) grows smaller, the defect density grows larger. The number of defects per square centimeter of resist is given by

$$D_0 = 1.5t^{-3}$$

2. The chip yield in fraction of good chips for each layer is given by

$$\eta = \frac{1}{1 + \alpha D_0 a}$$

where a is the active area of the chip. Assume that 50 percent of the defects are “fatal” defects ($\alpha = 0.5$) detected after manufacturing the chip.

Assume four layers are required for the device. The overall yield is based on a series formula:

$$\eta = \frac{1}{(1 + \alpha D_0 a)^4}$$

3. Throughput decreases as resist thickness increases. A typical relationship is

$$V(\text{wafers/h}) = 125 - 50t + 5t^2$$

Each wafer has 100 chip sites with 0.25 cm^2 active area. The daily production level is to be 2500 finished wafers. Find the resist thickness to be used to maximize the number of good chips per hour. Assume $0.5 \leq t \leq 2.5$ as the expected range. First use cubic interpolation to find the optimal value of t , t^* . How many parallel production lines are required for t^* , assuming 20 h/day operation each? How many iterations are needed to reach the optimum if you use quadratic interpolation?

Solution. The objective function to be maximized is the number of good chips per hour, which is found by multiplying the yield, the throughput, and the number of chips per wafer (= 100):

$$f = V\eta = (125 - 50t + 5t^2) \frac{100}{[1 + 0.5(1.5t^{-3})(0.25)]^4}$$

Using initial guesses of $t = 1.0$ and 2.0 , cubic interpolation yielded the following values of f :

t	f	f'
1.0	4023.05	5611.10
2.0	4101.73	-2170.89
1.414	4973.22	-148.70
1.395	4974.60	3.68 (optimum)

Because f is multiplied by 100, f' after two iterations is small enough. Figure E5.4 is a plot of the objective function $f(t)$.

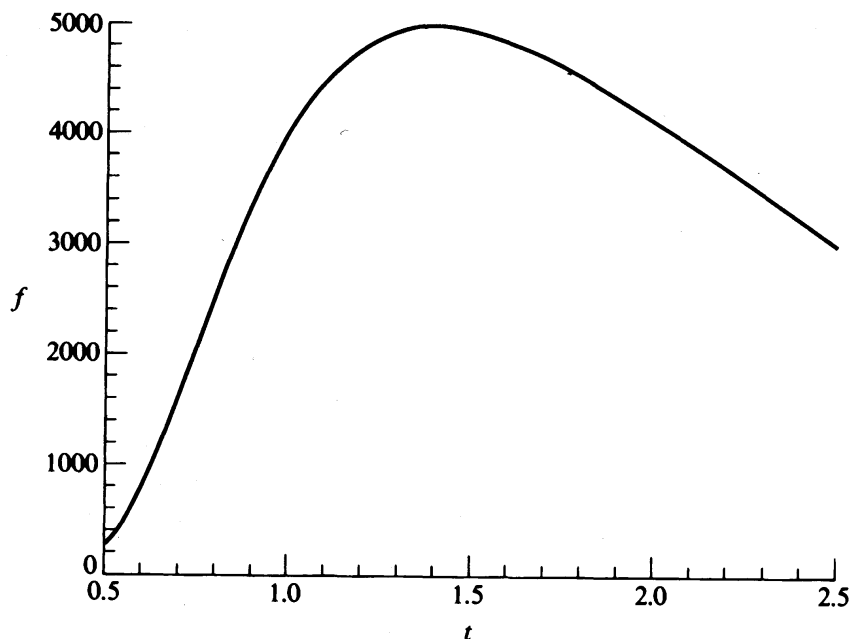


FIGURE E5.4

Plot of objective function (number of good chips per hour) versus resist thickness, $t(\mu\text{m})$.

The throughput for $t^* = 1.395$ is

$$V = 65.02 \text{ wafers/h}$$

If a production line is operated 20 h/day, two lines are needed to achieve 2500 wafers/day.

If quadratic interpolation is used with starting points of $t = 1, 2,$ and $3,$ the following iterative sequence results:

t	f	f'
1.0	4023.05	5611.10
2.0	4101.73	-2170.89
3.0	1945.40	-1891.73
1.535	4904.08	-942.28
1.511	4924.73	-810.91
1.434	4968.58	-304.19
1.420	4972.17	-196.10
1.406	4974.20	-81.98
1.401	4974.48	-44.78
1.398	4974.58	-20.24
1.397	4974.60	-10.76
1.396	4974.61	-5.01

5.5 HOW ONE-DIMENSIONAL SEARCH IS APPLIED IN A MULTIDIMENSIONAL PROBLEM

In minimizing a function $f(\mathbf{x})$ of several variables, the general procedure is to (a) calculate a search direction and (b) reduce the value of $f(\mathbf{x})$ by taking one or more steps in that search direction. Chapter 6 describes in detail how to select search directions. Here we explain how to take steps in the search direction as a function of a single variable, the step length α . The process of choosing α is called a *unidimensional search* or *line search*.

Examine Figure 5.7 in which contours of a function of two variables are displayed:

$$f(\mathbf{x}) = x_1^4 - 2x_2x_1^2 + x_2^2 + x_1^2 - 2x_1 + 5$$

Suppose that the negative gradient of $f(\mathbf{x})$, $-\nabla f(\mathbf{x})$, is selected as the search direction starting at the point $\mathbf{x}^T = [1 \ 2]$. The negative gradient is the direction that maximizes the rate of change of $f(\mathbf{x})$ in moving toward the minimum. To move in this direction we want to calculate a new \mathbf{x}

$$\mathbf{x}_{\text{new}} = \mathbf{x}_{\text{old}} + \alpha \mathbf{s}$$

where \mathbf{s} is the search direction, a vector, and α is a scalar denoting the distance moved along the search direction. Note $\alpha \mathbf{s} \equiv \Delta \mathbf{x}$, the vector for the step to be taken (encompassing both direction and distance).

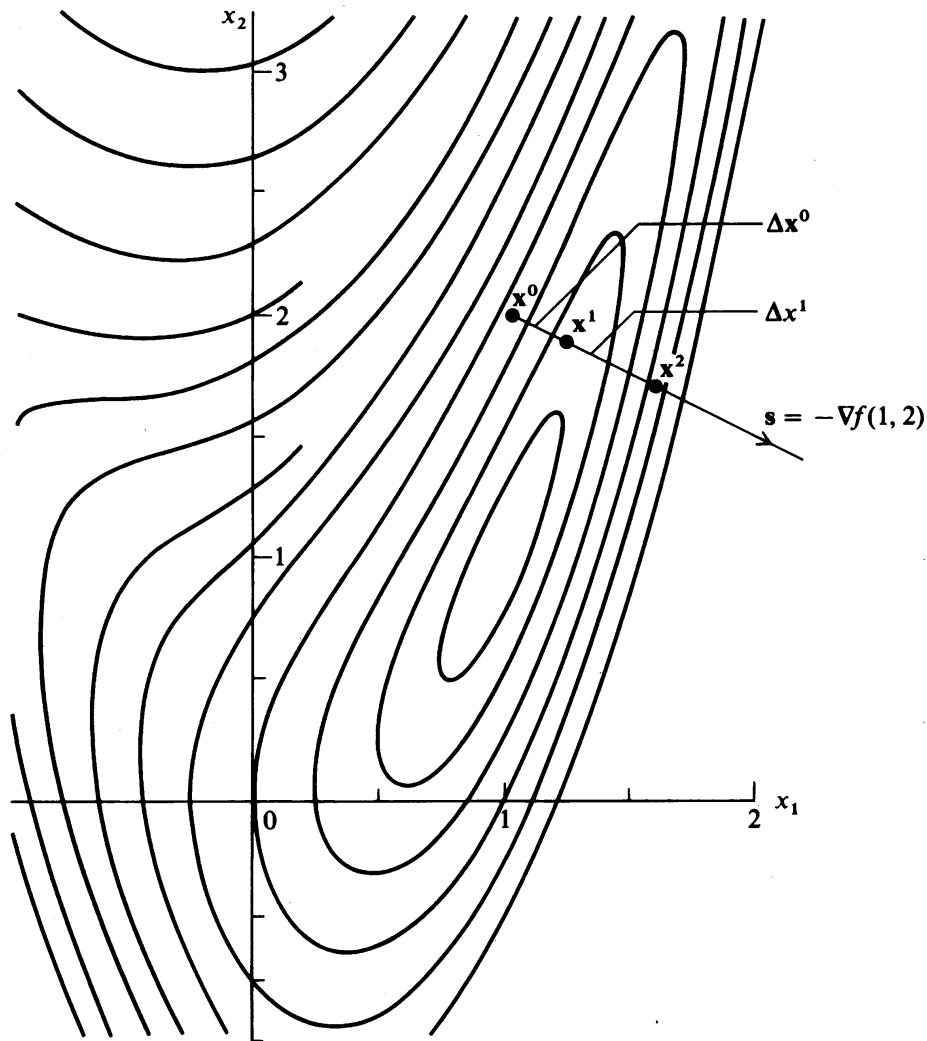


FIGURE 5.7
Unidimensional search to bracket the minimum.

Execution of a unidimensional search involves calculating a value of α and then taking steps in each of the coordinate directions as follows:

$$\text{In the } x_1 \text{ direction: } x_{1,\text{new}} = x_{1,\text{old}} + \alpha s_1$$

$$\text{In the } x_2 \text{ direction: } x_{2,\text{new}} = x_{2,\text{old}} + \alpha s_2$$

where s_1 and s_2 are the two components of s in the x_1 and x_2 directions, respectively. Repetition of this procedure accomplishes the unidimensional search.

EXAMPLE 5.5 EXECUTION OF A UNIDIMENSIONAL SEARCH

We illustrate two stages in bracketing the minimum in minimizing the function from Fox (1971)

$$f(\mathbf{x}) = x_1^4 - 2x_2x_1^2 + x_2^2 + x_1^2 - 2x_1 + 5$$

in the negative gradient direction

$$-\nabla f(\mathbf{x}) = -\begin{bmatrix} 4x_1^3 - 4x_2x_1 + 2x_1 - 2 \\ -2x_1^2 + 2x_2 \end{bmatrix}$$

starting at $\mathbf{x}^T = [1 \ 2]$ where $f(\mathbf{x}) = 5$. Here

$$\mathbf{s} = -\nabla f(1, 2) = -\begin{bmatrix} -4 \\ 2 \end{bmatrix}$$

We start to bracket the minimum by taking $\alpha^0 = 0.05$

$$x_1^1 = x_1^0 + (0.05)(4) = 1.2 \quad (a)$$

$$x_2^1 = x_2^0 + (0.05)(-2) = 1.9 \quad (b)$$

Steps (a) and (b) consist of one overall step in the direction $\mathbf{s} = [4 \ -2]^T$, and yield $\Delta\mathbf{x}^T = [0.2 \ -0.1]$. At \mathbf{x}^1 , $f(1.2, 1.9) = 4.25$, an improvement.

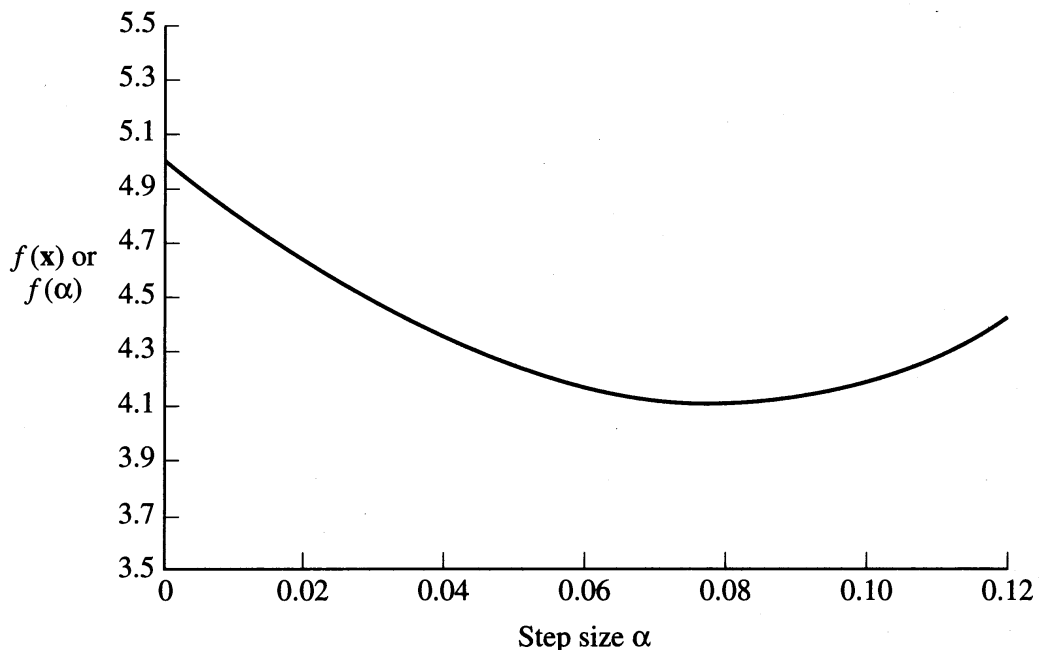


FIGURE E5.5

Values of $f(x)$ along the gradient vector $[4 \ -2]^T$ starting at $[1 \ 2]^T$.

For the next step, we let $\alpha^1 = 2\alpha^0 = 0.1$, and take another step in the same direction:

$$x_1^2 = x_1^1 + 0.1(4) = 1.6$$

$$x_2^2 = x_2^1 + 0.1(-2) = 1.7$$

$$\Delta \mathbf{x}^1 = [0.4 \quad -0.2]^T$$

At \mathbf{x}^2 , $f(1.6, 1.7) = 5.10$, so that the minimum of $f(x)$ in direction \mathbf{s} has been bracketed. Examine Figure 5.7. The optimal value of α along the search direction can be found to be $\tilde{\alpha}^* = 0.0797$ by one of the methods described in this chapter. Figure E5.5 shows a plot of f versus α along the search direction.

5.6 EVALUATION OF UNIDIMENSIONAL SEARCH METHODS

In this chapter we described and illustrated only a few unidimensional search methods. Refer to Luenberger (1984), Bazarra et al. (1993), or Nash and Sofer (1996) for many others. Naturally, you can ask which unidimensional search method is best to use, most robust, most efficient, and so on. Unfortunately, the various algorithms are problem-dependent even if used alone, and if used as subroutines in optimization codes, also depend on how well they mesh with the particular code. Most codes simply take one or a few steps in the search direction, or in more than one direction, with no requirement for accuracy—only that $f(x)$ be reduced by a sufficient amount.

REFERENCES

- Bazarra, M. S.; H. D. Sherali; and C. M. Shetty. *Nonlinear Programming: Theory and Algorithms*. Wiley, New York (1993).
- Becker, H. A.; P. L. Douglas; and S. Ilias. "Development of Optimization Strategies for Industrial Grain Dryer Systems." *Can J Chem Eng*, **62**: 738–745 (1984).
- Beveridge, G. S. G.; and R. S. Schechter. *Optimization: Theory and Practice*. McGraw-Hill, New York (1970).
- Cook, L. N. "Laboratory Approach Optimizes Filter-Aid Addition." *Chem Eng*, July 23, 1984: 45–50.
- Dennis, J. E.; and R. B. Schnabel. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. Prentice-Hall, Englewood Cliffs, New Jersey (1983) chapter 2.
- Fox, R. L. *Optimization Methods for Engineering Design*. Addison-Wesley, Reading, Massachusetts (1971) p. 42.
- Luenberger, D. G. *Linear and Nonlinear Programming*. 2nd ed. Addison-Wesley, Menlo Park, CA (1984).
- Nash, S. G.; and A. Sofer. *Linear and Nonlinear Programming*. McGraw-Hill, New York (1996).
- Wilde, D. J. *Optimum Seeking Methods*. Prentice-Hall, Englewood Cliffs, New Jersey (1964).
- Wilde, D. J. "Hidden Optima in Engineering Design." In *Constructive Approaches to Mathematical Models*. Academic Press, New York (1979): 243–248.

SUPPLEMENTARY REFERENCES

- Beightler, C. S.; D. T. Phillips; and D. J. Wild. *Foundations of Optimization*. 2nd ed. Prentice-Hall, Englewood Cliffs, New Jersey (1979).
- Brent, R. P. *Algorithms for Minimization Without Derivatives*. Prentice-Hall, Englewood Cliffs, New Jersey (1973).
- Cooper, L.; and D. Steinberg. *Introduction to Methods of Optimization*. W. B. Saunders Co., Philadelphia (1970).
- Reklaitis, G. V.; R. A. Ravindran; and K. M. Ragsdell. *Engineering Optimization*. Wiley-Interscience, New York (1983).
- Shoup, T. E.; and F. Mistree. *Optimization Methods with Applications for Personal Computers*. Prentice-Hall, Englewood Cliffs, New Jersey (1987).
- Weixnan, L. *Optimal Block Search*. Helderman, Berlin (1984).

PROBLEMS

- 5.1 Can you bracket the minimum of the following function

$$f(x) = e^x - 1.5x^2$$

starting at $x = 0$? Select different step sizes (small and large), and explain your results. If you have trouble in the analysis, you might plot the function.

- 5.2 Bracket the minimum of the following functions:

- (a) $f(x) = e^x + 1.5x^2$
- (b) $f(x) = 0.5(x^2 + 1)(x + 1)$
- (c) $f(x) = x^3 - 3x$
- (d) $f(x) = 2x^2(x - 2)(x + 2)$
- (e) $f(x) = 0.1x^6 - 0.29x^5 + 2.31x^4 - 8.33x^3 + 12.89x^2 - 6.8x + 1$

- 5.3 Minimize $f = (x - 1)^4$ via (a) Newton's method and (b) the quasi-Newton (secant) method, starting at (1) $x = -1$, (2) $x = -0.5$, and (3) $x = 0.0$.
- 5.4 Apply a sequential one-dimensional search technique to reduce the interval of uncertainty for the maximum of the function $f = 6.64 + 1.2x - x^2$ from $[0, 1]$ to less than 2 percent of its original size. Show all the iterations.
- 5.5 List three reasons why a quasi-Newton (secant) search for the minimum of a function of one variable will fail to find a local minimum.
- 5.6 Minimize the function $f = (x - 1)^4$. Use quadratic interpolation but no more than a maximum of ten function evaluations. The initial three points selected are $x_1 = 0$, $x_2 = 0.5$, and $x_3 = 2.0$.
- 5.7 Repeat Problem 5.6 but use cubic interpolation via function and derivative evaluations. Use $x_1 = 0.5$ and $x_2 = 2.0$ for a first guess.

5.8 Repeat Problem 5.6 for cubic interpolation with four function values: $x_1 = 1.5$, $x_2 = 3.0$, $x_3 = 4.0$, and $x_4 = 4.5$.

5.9 Carry out the initial and one additional stage of the numerical search for the minimum of

$$f(x) = 2x^3 - 5x^2 - 8 \quad x \geq 1$$

by (a) Newton's method (start at $x = 1$), (b) the quasi-Newton (secant) method (pick a starting point), and (c) polynomial approximation (pick starting points including $x = 1$).

5.10 Find the maximum of the following function

$$f(x) = 1 - 8x + 2x^2 - \frac{10}{3}x^3 + \frac{1}{4}x^4 + \frac{4}{5}x^5 - \frac{1}{6}x^6$$

$$\text{Hint: } f'(x) = (1 + x)^2(2 - x)^3$$

(a) Analytically. (b) By Newton's method (two iterations will suffice). Start at $x = -2$. List each step of the procedure. (c) By quadratic interpolation (two iterations will suffice). Start at $x = -2$. List each step of the procedure.

5.11 Determine the relative rates of convergence for (1) Newton's method, (2) a finite difference Newton method, (3) quasi-Newton method, (4) quadratic interpolation, and (5) cubic interpolation, in minimizing the following functions:

(a) $x^2 - 6x + 3$ (b) $\sin(x)$ with $0 < x < 2\pi$ (c) $x^4 - 20x^3 + 0.1x$

5.12 The total annual cost of operating a pump and motor C in a particular piece of equipment is a function of x , the size (horsepower) of the motor, namely

$$C = \$500 + \$0.9x + \frac{\$0.03}{x}(150,000)$$

Find the motor size that minimizes the total annual cost.

5.13 A boiler house contains five coal-fired boilers, each with a nominal rating of 300 boiler horsepower (BHP). If economically justified, each boiler can be operated at a rating of 350 percent of nominal. Due to the growth of manufacturing departments, it has become necessary to install additional boilers. Refer to the following data. Determine the percent of nominal rating at which the present boilers should be operated. *Hint:* Minimize total costs per year BHP output.

Data: The cost of fuel, coal, including the cost of handling coal and removing cinders, is \$7 per ton, and the coal has a heating value of 14,000 Btu/lb. The overall efficiency of the boilers, from coal to steam, has been determined from tests of the present boilers operated at various ratings as:

Percent of nominal rating, R	Percent overall thermal efficiency, E
100	75
150	76
200	74
225	72
250	69
275	65
300	61

The annual fixed charges C_F in dollars per year on each boiler are given by the equation:

$$C_F = 14,000 + 0.04R^2$$

Assume 8550 hours of operation per year.

Hint: You will find it helpful to first obtain a relation between R and E by least squares (refer to Chapter 2) to eliminate the variable E .

- 5.14 A laboratory filtration study is to be carried out at constant rate. The basic equation (Cook, 1984) comes from the relation

$$\text{Flow-rate} \propto \frac{(\text{Pressure drop})(\text{Filter area})}{(\text{Fluid viscosity})(\text{Cake thickness})}$$

Cook expressed filtration time as

$$t_f = \beta \frac{\Delta P_c A^2}{\mu M^2 c} x_c \exp(-ax_c + b)$$

where t_f = time to build up filter cake, min

ΔP_c = pressure drop across cake, psig (20)

A = filtration area, ft² (250)

μ = filtrate viscosity, centipoise (20)

M = mass flow of filtrate, lb_m/min (75)

c = solids concentration in feed to filter, lb_m/lb_m filtrate (0.01)

x_c = mass fraction solids in dry cake

a = constant relating cake resistance to solids fraction (3.643)

h = constant relating cake resistance to solids fraction (2.680)

$\beta = 3.2 \times 10^{-8}$ (lb_m/ft)²

Numerical values for each parameter are given in parentheses. Obtain the maximum time for filtration as a function of x_c by a numerical unidimensional search.

- 5.15 An industrial dryer for granular material can be modeled (Becker et al., 1984) with the total specific cost of drying C (\$/m³) being

$$C = [1.767 \ln(W_0/W_D)/\beta V_i] \frac{(F_A C_{pA} + UA)\Delta T C'_p}{\Delta H_c + PC'_E + C'_L}$$

where A = heat transfer area of dryer normal to the air flow, m² (153.84)

β = constant, function of air plenum temperature and initial moisture level

C'_E = unit cost of electricity, \$/kWh (0.0253)

C'_L = unit cost of labor, \$/h (15)

C'_p = unit cost of propane, \$/kg (0.18)

C_{pA} = specific heat of air, J/kg K (1046.75)

F_A = flow-rate of air, kg/h (3.38×10^5)

ΔH_c = heat combustion of propane, J/kg (4.64×10^7)

P = electrical power, kW (188)

ΔT = temperature difference ($T - T_1$), K; the plenum air temperature T minus the inlet air temperature T_1 ($T_1 = 390$ K)

U = overall heat transfer coefficient from dryer to atmosphere,
 $W/(m^2)(K)(45)$

V_t = total volume of the dryer, m^3 (56)

W_D = final grain moisture content (dry basis), kg/kg (0.1765)

W_0 = initial moisture content (dry basis), kg/kg (0.500)

Numerical values for each parameter are given in parentheses. Values for the coefficient are given by

$$\beta = (-0.2631125 + 0.0028958T) W_0^{(-0.2368125 + 0.0009667T)}$$

Find the minimum cost as a function of the plenum temperature T (in kelvin).

5.16 The following is an example from D. J. Wilde (1979).

The first example was formulated by Stoecker* to illustrate the steepest descent (gradient) direct search method. It is proposed to attach a vapor recondensation refrigeration system to lower the temperature, and consequently vapor pressure, of liquid ammonia stored in a steel pressure vessel, for this would permit thinner vessel walls. The tank cost saving must be traded off against the refrigeration and thermal insulation cost to find the temperature and insulation thickness minimizing the total annual cost. Stoecker showed the total cost to be the sum of insulation cost $i \equiv 400x^{0.9}$ (x is the insulation thickness, in.), the vessel cost $v \equiv 1000 + 22(p - 14.7)^{1.2}$ (p is the absolute pressure, psia), and the recondensation cost $r \equiv 144(80 - t)/x$ (t is the temperature, °F). The pressure is related to the temperature by

$$\ln p = -3950(t - 460)^{-1} + 11.86$$

By direct gradient search, iterated 16 times from a starting temperature of 50°F, the total annual cost is found to have a local minimum at $x = 5.94$ in. and $t = 6.29$ °F, where the cost is \$53,400/yr. The reader can verify, however, that an ambient system (80°F) without any recondensation only costs \$52,000/yr, a saving of 3%.

Is the comment in the example true?

*Stoecker, W. F. In "Design of Thermal Systems." McGraw-Hill, New York (1971), pp. 152–155.

6

UNCONSTRAINED MULTIVARIABLE OPTIMIZATION

6.1 Methods Using Function Values Only	183
6.2 Methods That Use First Derivatives	189
6.3 Newton's Method	197
6.4 Quasi-Newton Methods	208
References	210
Supplementary References	211
Problems	211

THE NUMERICAL OPTIMIZATION of general nonlinear multivariable objective functions requires efficient and robust techniques. Efficiency is important because these problems require an iterative solution procedure, and trial and error becomes impractical for more than three or four variables. Robustness (the ability to achieve a solution) is desirable because a general nonlinear function is unpredictable in its behavior; there may be relative maxima or minima, saddle points, regions of convexity, concavity, and so on. In some regions the optimization algorithm may progress very slowly toward the optimum, requiring excessive computer time. Fortunately, we can draw on extensive experience in testing nonlinear programming algorithms for unconstrained functions to evaluate various approaches proposed for the optimization of such functions.

In this chapter we discuss the solution of the unconstrained optimization problem:

Find: $\mathbf{x}^* = [x_1^* \ x_2^* \ \cdots \ x_n^*]^T$ that minimizes $f(x_1, x_2, \dots, x_n) \equiv f(\mathbf{x})$

Most effective iterative procedures alternate between two phases in the optimization. At iteration k , where the current \mathbf{x} is \mathbf{x}^k , they do the following:

1. Choose a search direction \mathbf{s}^k
2. Minimize along that direction (usually inexactly) to find a new point

$$\mathbf{x}^{k+1} = \mathbf{x}^k + \alpha^k \mathbf{s}^k$$

where α^k is a positive scalar called the step size. The step size is determined by an optimization process called a line search as described in Chapter 5.

In addition to 1 and 2, an algorithm must specify

3. The initial starting vector $\mathbf{x}^0 = [x_1^0 \ x_2^0 \ \cdots \ x_n^0]^T$ and
4. The convergence criteria for termination.

From a given starting point, a search direction is determined, and $f(\mathbf{x})$ is minimized in that direction. The search stops based on some criteria, and then a new search direction is determined, followed by another line search. The line search can be carried out to various degrees of precision. For example, we could use a simple successive doubling of the step size as a screening method until we detect the optimum has been bracketed. At this point the screening search can be terminated and a more sophisticated method employed to yield a higher degree of accuracy. In any event, refer to the techniques discussed in Chapter 5 for ways to carry out the line search.

The NLP (nonlinear programming) methods to be discussed in this chapter differ mainly in how they generate the search directions. Some nonlinear programming methods require information about derivative values, whereas others do not use derivatives and rely solely on function evaluations. Furthermore, finite difference substitutes can be used in lieu of derivatives as explained in Section 8.10. For differentiable functions, methods that use analytical derivatives almost always use less computation time and are more accurate, even if finite difference approxima-

tions are used. Symbolic codes can be employed to obtain analytical derivatives but this may require more computer time than finite differencing to get derivatives. For nonsmooth functions, a function-values-only method may be more successful than using a derivative-based method. We first describe some simple nonderivative methods and then present a series of methods that use derivative information. We also show how the nature of the objective function influences the effectiveness of the particular optimization algorithm.

6.1 METHODS USING FUNCTION VALUES ONLY

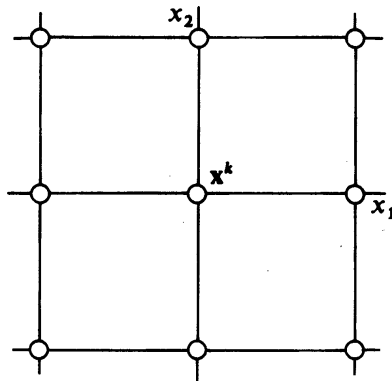
Some methods do not require the use of derivatives in determining the search direction. Under some circumstances the methods described in this section can be used effectively, but they may be inefficient compared with methods discussed in subsequent sections. They have the advantage of being simple to understand and execute.

6.1.1 Random Search

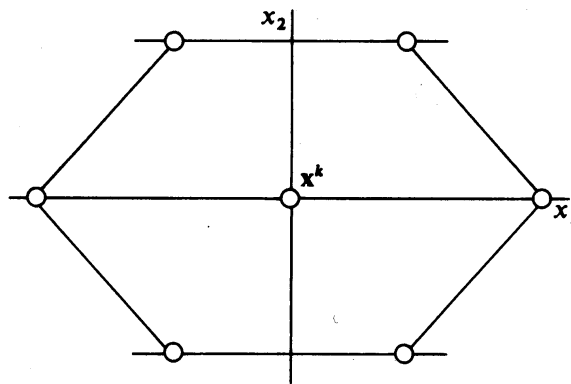
A random search method simply selects a starting vector \mathbf{x}^0 , evaluates $f(\mathbf{x})$ at \mathbf{x}^0 , and then randomly selects another vector \mathbf{x}^1 and evaluates $f(\mathbf{x})$ at \mathbf{x}^1 . In effect, both a search direction and step length are chosen simultaneously. After one or more stages, the value of $f(\mathbf{x}^k)$ is compared with the best previous value of $f(\mathbf{x})$ from among the previous stages, and the decision is made to continue or terminate the procedure. Variations of this form of random search involve randomly selecting a search direction and then minimizing (possibly by random steps) in that search direction as a series of cycles. Clearly, *the* optimal solution can be obtained with a probability of 1 only as $k \rightarrow \infty$ but as a practical matter, if the objective function is quite flat, a suboptimal solution may be quite acceptable. Even though the method is inefficient insofar as function evaluations are concerned, it may provide a good starting point for another method. You might view random search as an extension of the case study method. Refer to Dixon and James (1980) for some practical algorithms.

6.1.2 Grid Search

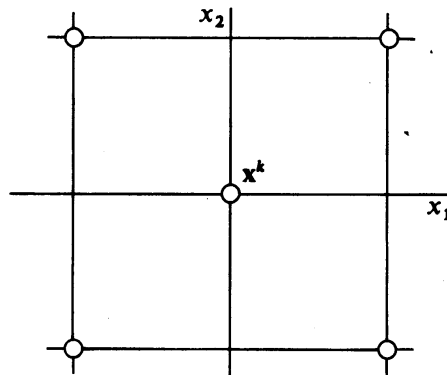
Methods of experimental design discussed in most basic statistics books can be applied equally well to minimizing $f(\mathbf{x})$ (see Chapter 2). You evaluate a series of points about a reference point selected according to some type of design such as the ones shown in Figure 6.1 (for an objective function of two variables). Next you move to the point that improves the objective function the most, and repeat.



(a) Three-level factorial design ($3^2 - 1 = 8$ points plus center)



(b) Hexagon design (6 points + center)

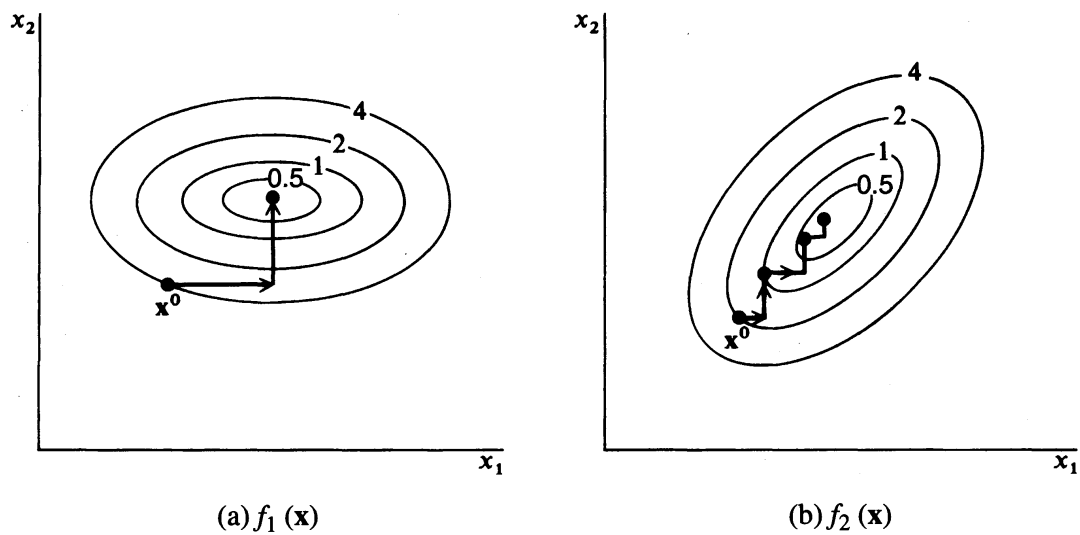


(c) Two-level factorial design ($2^2 = 4$ points plus center)

FIGURE 6.1

Various grid search designs to select vectors \mathbf{x} to evaluate $f(\mathbf{x})$.

For $n = 30$, we must examine $3^{30} - 1 = 2.0588 \times 10^{14}$ values of $f(\mathbf{x})$ if a three-level factorial design is to be used, obviously a prohibitive number of function evaluations.

**FIGURE 6.2**

Execution of a univariate search on two different quadratic functions.

6.1.3 Univariate Search

Another simple optimization technique is to select n fixed search directions (usually the coordinate axes) for an objective function of n variables. Then $f(\mathbf{x})$ is minimized in each search direction sequentially using a one-dimensional search. This method is effective for a quadratic function of the form

$$f_1(\mathbf{x}) = \sum_{i=1}^n c_i x_i^2$$

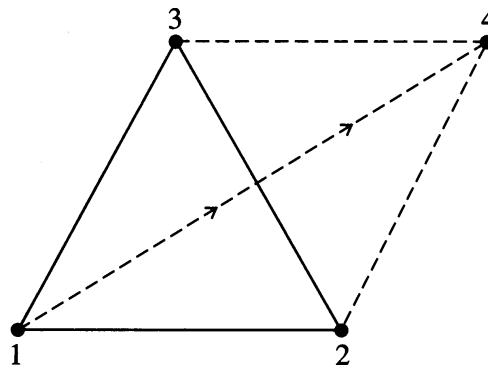
because the search directions line up with the principal axes as indicated in Figure 6.2a. However, it does not perform satisfactorily for more general quadratic objective functions of the form

$$f_2(\mathbf{x}) = \sum_{i=1}^n \sum_{j=1}^n d_{ij} x_i x_j$$

as illustrated in Figure 6.2b. For the latter case, the changes in \mathbf{x} decrease as the optimum is neared, so many iterations will be required to attain high accuracy.

6.1.4 Simplex Search Method

The method of the “Sequential Simplex” formulated by Spendley, Hext, and Himsworth (1962) selects points at the vertices of the simplex at which to evaluate $f(\mathbf{x})$. In two dimensions the figure is an equilateral triangle. Examine Figure 6.3. In three dimensions this figure becomes a regular tetrahedron, and so on. Each search direction points away from the vertex having the highest value of $f(\mathbf{x})$ to the other vertices in the simplex. Thus, the direction of search changes, but the step size is

**FIGURE 6.3**

Reflection to a new point in the simplex method.
At point 1, $f(\mathbf{x})$ is greater than f at points 2 or 3.

fixed for a given size simplex. Let us use a function of two variables to illustrate the procedure.

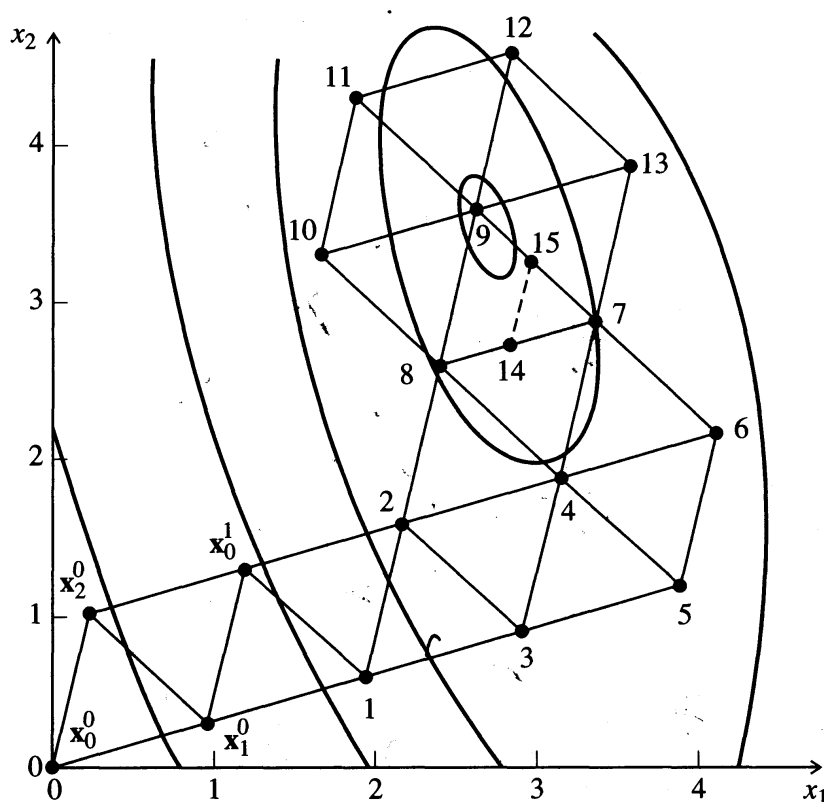
At each iteration, to minimize $f(\mathbf{x})$, $f(\mathbf{x})$ is evaluated at each of three vertices of the triangle. The direction of search is oriented away from the point with the highest value for the function through the centroid of the simplex. By making the search direction bisect the line between the other two points of the triangle, the direction goes through the centroid. A new point is selected in this reflected direction (as shown in Figure 6.3), preserving the geometric shape. The objective function is then evaluated at the new point, and a new search direction is determined. The method proceeds, rejecting one vertex at a time until the simplex straddles the optimum. Various rules are used to prevent excessive repetition of the same cycle or simplexes.

As the optimum is approached, the last equilateral triangle straddles the optimum point or is within a distance of the order of its own size from the optimum (examine Figure 6.4). The procedure cannot therefore get closer to the optimum and repeats itself so that the simplex size must be reduced, such as halving the length of all the sides of the simplex containing the vertex where the oscillation started. A new simplex composed of the midpoints of the ending simplex is constructed. When the simplex size is smaller than a prescribed tolerance, the routine is stopped. Thus, the optimum position is determined to within a tolerance influenced by the size of the simplex.

Nelder and Mead (1965) described a more efficient (but more complex) version of the simplex method that permitted the geometric figures to expand and contract continuously during the search. Their method minimized a function of n variables using $(n + 1)$ vertices of a flexible polyhedron. Details of the method together with a computer code to execute the algorithm can be found in Avriel (1976).

6.1.5 Conjugate Search Directions

Experience has shown that conjugate directions are much more effective as search directions than arbitrarily chosen search directions, such as in univariate search, or

**FIGURE 6.4**

Progression to the vicinity of the optimum and oscillation around the optimum using the simplex method of search. The original vertices are x_0^0 , x_1^0 , and x_2^0 . The next point (vertex) is x_0^1 . Succeeding new vertices are numbered starting with 1 and continuing to 13 at which point a cycle starts to repeat. The size of the simplex is reduced to the triangle determined by points 7, 14, and 15; and then the procedure is continued (not shown).

even orthogonal search directions. Two directions s^i and s^j are said to be *conjugate* with respect to a positive-definite matrix \mathbf{Q} if

$$(s^i)^T \mathbf{Q} (s^j) = 0 \quad (6.1)$$

In general, a set of n linearly independent directions of search s^0, s^1, \dots, s^{n-1} are said to be conjugate with respect to a positive-definite square matrix \mathbf{Q} if

$$(s^i)^T \mathbf{Q} s^j = 0 \quad 0 \leq i \neq j \leq n - 1 \quad (6.2)$$

In optimization the matrix \mathbf{Q} is the Hessian matrix of the objective function, \mathbf{H} . For a *quadratic function* $f(\mathbf{x})$ of n variables, in which \mathbf{H} is a constant matrix, you are guaranteed to reach the minimum of $f(\mathbf{x})$ in n stages if you minimize *exactly* on each stage (Dennis and Schnabel, 1996). In n dimensions, many different sets of conjugate directions exist for a given matrix \mathbf{Q} . In two dimensions, however, if you choose an initial direction s^1 and \mathbf{Q} , s^2 is fully specified as illustrated in Example 6.1.

Orthogonality is a special case of conjugacy because when $\mathbf{Q} = \mathbf{I}$, $(\mathbf{s}^i)^T \mathbf{s}^j = 0$ in Equation (6.2). If the coordinates of \mathbf{x} are translated and rotated by suitable transformations so as to align the new principal axes of $\mathbf{H}(\mathbf{x})$ with the eigenvectors of $\mathbf{H}(\mathbf{x})$ and to place the center of the coordinate system at the stationary point of $f(\mathbf{x})$ (refer to Figures 4.12 through 4.15), then conjugacy can be interpreted as orthogonality in the space of the transformed coordinates.

Although authors and practitioners refer to a class of unconstrained optimization methods as “methods that use conjugate directions,” for a general nonlinear function, the conjugate directions exist only for a quadratic approximation of the function at a single stage k . Once the objective function is modeled by a new approximation at stage $(k + 1)$, the directions on stage k are unlikely to be conjugate to any of the directions selected in stage $(k + 1)$.

EXAMPLE 6.1 CALCULATION OF CONJUGATE DIRECTIONS

Suppose we want to minimize $f(\mathbf{x}) = 2x_1^2 + x_2^2 - 3$ starting at $(\mathbf{x}^0)^T = [1 \ 1]$ with the initial direction being $\mathbf{s}^0 = [-4 \ -2]^T$. Find a conjugate direction to the initial direction \mathbf{s}^0 .

Solution

$$\mathbf{s}^0 = - \begin{bmatrix} 4 \\ 2 \end{bmatrix} \quad \mathbf{H}(\mathbf{x}) = \begin{bmatrix} 4 & 0 \\ 0 & 2 \end{bmatrix}$$

We need to solve Equation (6.2) for $\mathbf{s}^1 = [s_1^1 \ s_2^1]^T$ with $\mathbf{Q} = \mathbf{H}$ and $\mathbf{s}^0 = [-4 \ -2]^T$.

$$(-1) \begin{bmatrix} 4 & 2 \end{bmatrix} \begin{bmatrix} 4 & 0 \\ 0 & 2 \end{bmatrix} \begin{bmatrix} s_1^1 \\ s_2^1 \end{bmatrix} = 0$$

Because s_1^1 is not unique, we can pick $s_1^1 = 1$ and determine s_2^1

$$\begin{bmatrix} -16 & -4 \end{bmatrix} \begin{bmatrix} 1 \\ s_2^1 \end{bmatrix} = 0$$

Thus $\mathbf{s}^1 = [1 \ -4]^T$ is a direction conjugate to $\mathbf{s}^0 = [-4 \ -2]^T$.

We can reach the minimum of $f(\mathbf{x})$ in two stages using first \mathbf{s}^0 and then \mathbf{s}^1 . Can we use the search directions in reverse order? From $\mathbf{x}^0 = [1 \ 1]^T$ we can carry out a numerical search in the direction $\mathbf{s}^0 = [-4 \ -2]^T$ to reach the point \mathbf{x}^1 . Quadratic interpolation can obtain the exact optimal step length because f is quadratic, yielding $\alpha = 0.27778$. Then

$$\mathbf{x}^1 = \mathbf{x}^0 - \alpha^0 \mathbf{s}^0 = \begin{bmatrix} 1 \\ 1 \end{bmatrix} + 0.27778 \begin{bmatrix} -4 \\ -2 \end{bmatrix} = \begin{bmatrix} -0.1111 \\ 0.4444 \end{bmatrix}$$

For the next stage, the search direction is $\mathbf{s}^1 = [1 \ -4]^T$, and the optimal step length calculated by quadratic interpolation is $\alpha^1 = 0.1111$. Hence

$$\mathbf{x}^2 = \mathbf{x}^1 + \alpha^1 \mathbf{s}^1 = \begin{bmatrix} -0.1111 \\ 0.4444 \end{bmatrix} + 0.1111 \begin{bmatrix} 1 \\ -4 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

as expected.

6.1.6 Summary

As mentioned earlier, nonlinear objective functions are sometimes nonsmooth due to the presence of functions like abs, min, max, or if-then-else statements, which can cause derivatives, or the function itself, to be discontinuous at some points. Unconstrained optimization methods that do not use derivatives are often able to solve nonsmooth NLP problems, whereas methods that use derivatives can fail. Methods employing derivatives can get “stuck” at a point of discontinuity, but the function-value-only methods are less affected. For smooth functions, however, methods that use derivatives are both more accurate and faster, and their advantage grows as the number of decision variables increases. Hence, we now turn our attention to unconstrained optimization methods that use only first partial derivatives of the objective function.

6.2 METHODS THAT USE FIRST DERIVATIVES

A good search direction should reduce (for minimization) the objective function so that if \mathbf{x}^0 is the original point and \mathbf{x}^1 is the new point

$$f(\mathbf{x}^1) < f(\mathbf{x}^0)$$

Such a direction \mathbf{s} is called a descent direction and satisfies the following requirement at any point

$$\nabla^T f(\mathbf{x}) \mathbf{s} < 0$$

To see why, examine the two vectors $\nabla f(\mathbf{x}^k)$ and \mathbf{s}^k in Figure 6.5. The angle between them is θ , hence

$$\nabla^T f(\mathbf{x}) \mathbf{s}^k = |\nabla f(\mathbf{x}^k)| |\mathbf{s}^k| \cos \theta$$

If $\theta = 90^\circ$ as in Figure 6.5, then steps along \mathbf{s}^k do not reduce (improve) the value of $f(\mathbf{x})$. If $0 \leq \theta < 90^\circ$, no improvement is possible and $f(\mathbf{x})$ increases. Only if $\theta > 90^\circ$ does the search direction yield smaller values of $f(\mathbf{x})$, hence $\nabla^T f(\mathbf{x}^k) \mathbf{s}^k < 0$.

We first examine the classic steepest descent method of using the gradient and then examine a conjugate gradient method.

6.2.1 Steepest Descent

The gradient is the vector at a point \mathbf{x} that gives the (local) direction of the greatest rate of increase in $f(\mathbf{x})$. It is orthogonal to the contour of $f(\mathbf{x})$ at \mathbf{x} . For maximization, the search direction is simply the gradient (when used the algorithm is called “steepest ascent”); for minimization, the search direction is the negative of the gradient (“steepest descent”)

$$\mathbf{s}^k = -\nabla f(\mathbf{x}^k) \quad (6.3)$$

In steepest descent at the k th stage, the transition from the current point \mathbf{x}^k to the new point \mathbf{x}^{k+1} is given by the following expression:

$$\mathbf{x}^{k+1} = \mathbf{x}^k + \Delta\mathbf{x}^k = \mathbf{x}^k + \alpha^k \mathbf{s}^k = \mathbf{x}^k - \alpha^k \nabla f(\mathbf{x}^k) \quad (6.4)$$

where $\Delta\mathbf{x}^k$ = vector from \mathbf{x}^k to \mathbf{x}^{k+1}

\mathbf{s}^k = search direction, the direction of steepest descent

α^k = scalar that determines the step length in direction \mathbf{s}^k

The negative of the gradient gives the direction for minimization but not the magnitude of the step to be taken, so that various steepest descent procedures are possible

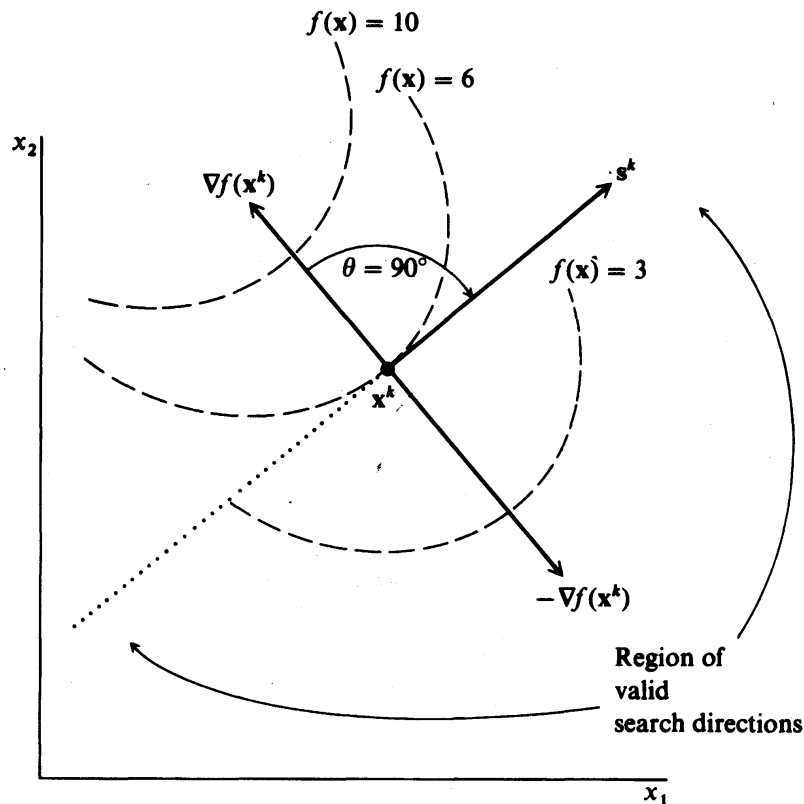


FIGURE 6.5

Identification of the region of possible search directions.

sible, depending on the choice of α^k . We assume that the value of $f(\mathbf{x})$ is continuously reduced. Because one step in the direction of steepest descent will not, in general, arrive at the minimum of $f(\mathbf{x})$, Equation (6.4) must be applied repetitively until the minimum is reached. At the minimum, the value of the elements of the gradient vector will each be equal to zero.

The step size α^k is determined by a line search, using methods like those described in Chapter 5. Although inexact line searches (not continued to the exact minimum) are always used in practice, insight is gained by examining the behavior of steepest descent when an exact line search is used.

First, let us consider the perfectly scaled quadratic objective function $f(\mathbf{x}) = x_1^2 + x_2^2$, whose contours are concentric circles as shown in Figure 6.6. Suppose we calculate the gradient at the point $\mathbf{x}^T = [2 \ 2]$

$$\nabla f(\mathbf{x}) = \begin{bmatrix} 2x_1 \\ 2x_2 \end{bmatrix} \quad \nabla f(2,2) = \begin{bmatrix} 4 \\ 4 \end{bmatrix} \quad \mathbf{H}(\mathbf{x}) = \mathbf{H} = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$$

The direction of steepest descent is

$$\mathbf{s} = - \begin{bmatrix} 4 \\ 4 \end{bmatrix}$$

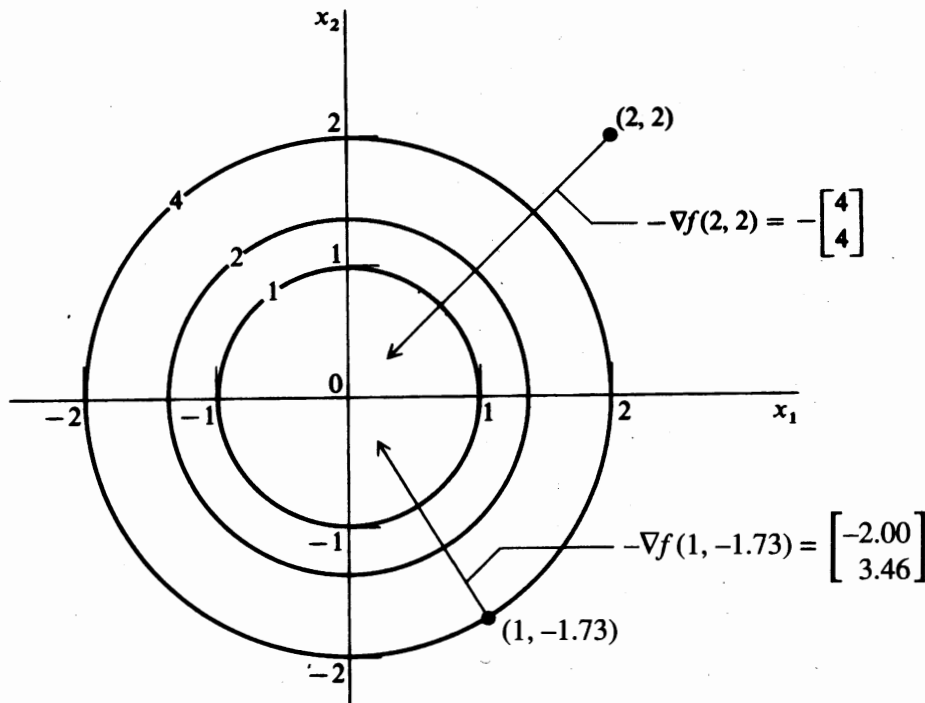
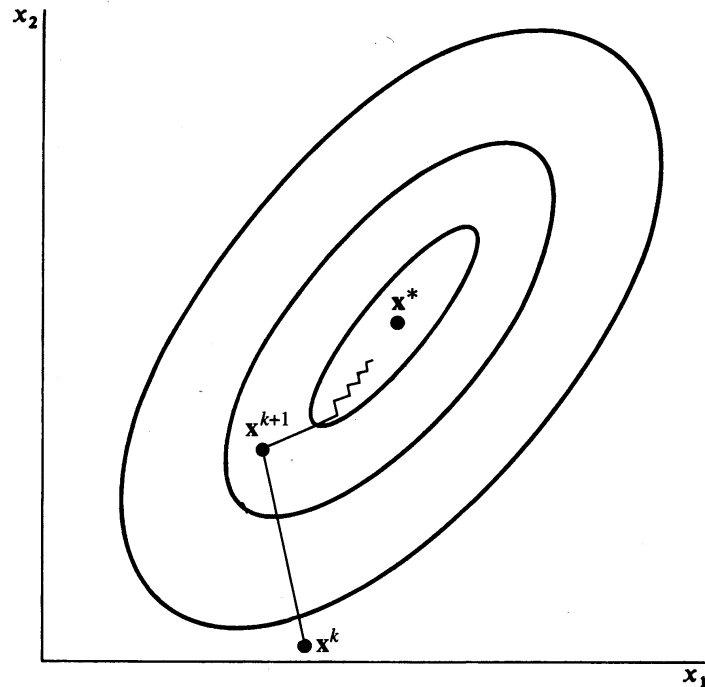


FIGURE 6.6

Gradient vector for $f(\mathbf{x}) = x_1^2 + x_2^2$.

**FIGURE 6.7**

Steepest descent method for a general quadratic function.

Observe that \mathbf{s} is a vector pointing toward the optimum at $(0, 0)$. In fact, the gradient at any point passes through the origin (the optimum).

On the other hand, for functions not so nicely scaled and that have nonzero off-diagonal terms in the Hessian matrix (corresponding to interaction terms such as x_1x_2), then the negative gradient direction is unlikely to pass directly through the optimum. Figure 6.7 illustrates the contours of a quadratic function of two variables that includes an interaction term. Observe that contours are tilted with respect to the axes. Interaction terms plus poor scaling corresponding to narrow valleys, or ridges, cause the gradient method to exhibit slow convergence.

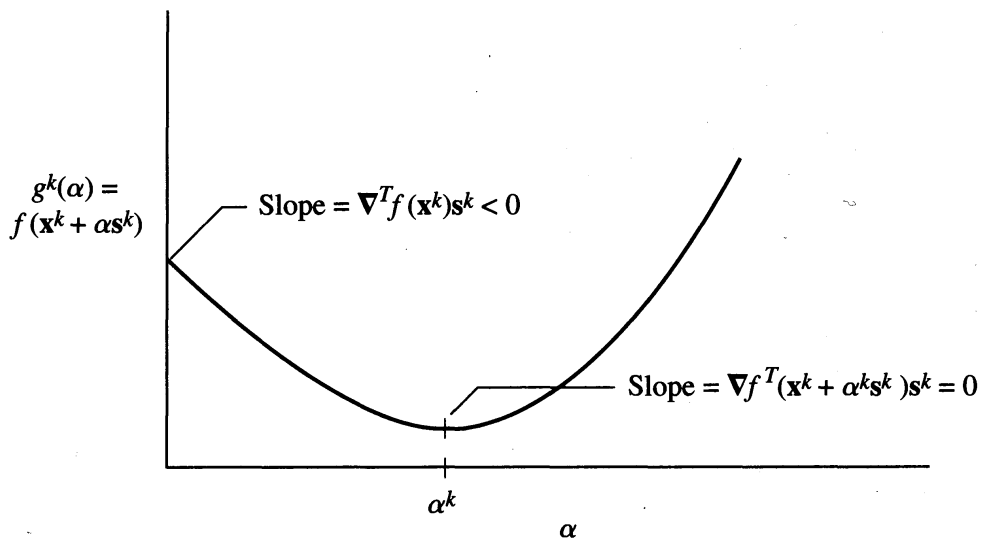
If α^k is chosen to minimize $f(\mathbf{x}^k + \alpha\mathbf{s}^k)$ exactly then at the minimum,

$$\frac{d}{d\alpha}f(\mathbf{x}^k + \alpha\mathbf{s}^k) = 0$$

We illustrate this in Figure 6.8 using the notation

$$g^k(\alpha) = f(\mathbf{x}^k + \alpha\mathbf{s}^k)$$

where g^k is the function value along the search direction for a given value of α . Because \mathbf{x}^k and \mathbf{s}^k are fixed at known values, g^k depends only on the step size α . If \mathbf{s}^k is a descent direction, then we can always find a positive α that causes f to decrease.

**FIGURE 6.8**

Exact line search along the search direction \mathbf{s}^k .

Using the chain rule

$$\begin{aligned} \frac{d}{d\alpha} f(\mathbf{x}^k + \alpha \mathbf{s}^k) &= \sum_i \frac{\partial f(\mathbf{x}^k + \alpha \mathbf{s}^k)}{\partial \mathbf{x}_i} s_i^k \\ &= (\mathbf{s}^k)^T \nabla f(\mathbf{x}^k + \alpha \mathbf{s}^k) \end{aligned}$$

In an exact line search, we choose α^k as the α that minimizes $g^k(\alpha)$, so

$$\left. \frac{dg^k}{d\alpha} \right|_{\alpha^k} = (\mathbf{s}^k)^T \nabla f(\mathbf{x}^k + \alpha \mathbf{s}^k) \Big|_{\alpha^k} = 0 \quad (6.5)$$

as shown in Figure 6.8. But when the inner product of two vectors is zero, the vectors are orthogonal, so if an exact line search is used, the gradient at the new point \mathbf{x}^{k+1} is orthogonal to the search direction \mathbf{s}^k . In steepest descent $\mathbf{s}^k = -\nabla f(\mathbf{x}^k)$, so the gradients at points \mathbf{x}^k and \mathbf{x}^{k+1} are orthogonal. This is illustrated in Figure 6.7, which shows that the orthogonality of successive search directions leads to a very inefficient zigzagging behavior. Although large steps are taken in early iterations, the step sizes shrink rapidly, and converging to an accurate solution of the optimization problem takes many iterations.

The steepest descent algorithm can be summarized in the following steps:

1. Choose an initial or starting point \mathbf{x}^0 . Thereafter at the point \mathbf{x}^k :
2. Calculate (analytically or numerically) the partial derivatives

$$\frac{\partial f(\mathbf{x})}{\partial x_j} \quad j = 1, \dots, n$$

3. Calculate the search vector

$$\mathbf{s}^k = -\nabla f(\mathbf{x}^k)$$

4. Use the relation

$$\mathbf{x}^{k+1} = \mathbf{x}^k + \alpha^k \mathbf{s}^k$$

to obtain the value of \mathbf{x}^{k+1} . To get α^k minimize $g^k(\alpha)$ numerically, as described in Chapter 5.

5. Compare $f(\mathbf{x}^{k+1})$ with $f(\mathbf{x}^k)$: if the change in $f(\mathbf{x})$ is smaller than some tolerance, stop. If not, return to step 2 and set $k = k + 1$. Termination can also be specified by stipulating some tolerance on the norm of $\nabla f(\mathbf{x}^k)$.

Steepest descent can terminate at any type of stationary point, that is, at any point where the elements of the gradient of $f(\mathbf{x})$ are zero. Thus you must ascertain if the presumed minimum is indeed a local minimum (i.e., a solution) or a saddle point. If it is a saddle point, it is necessary to employ a nongradient method to move away from the point, after which the minimization may continue as before. The stationary point may be tested by examining the Hessian matrix of the objective function as described in Chapter 4. If the Hessian matrix is not positive-definite, the stationary point is a saddle point. Perturbation from the stationary point followed by optimization should lead to a local minimum \mathbf{x}^* .

The basic difficulty with the steepest descent method is that it is too sensitive to the scaling of $f(\mathbf{x})$, so that convergence is very slow and what amounts to oscillation in the \mathbf{x} space can easily occur. For these reasons steepest descent or ascent is not a very effective optimization technique. Fortunately, conjugate gradient methods are much faster and more accurate.

6.2.2 Conjugate Gradient Methods

The earliest conjugate gradient method was devised by Fletcher and Reeves (1964). If $f(\mathbf{x})$ is quadratic and is minimized exactly in each search direction, it has the desirable features of converging in at most n iterations because its search directions are conjugate. The method represents a major improvement over steepest descent with only a marginal increase in computational effort. It combines current information about the gradient vector with that of gradient vectors from previous iterations (a memory feature) to obtain the new search direction. You compute the search direction by a linear combination of the current gradient and the previous search direction. The main advantage of this method is that it requires only a small amount of information to be stored at each stage of calculation and thus can be applied to very large problems. The steps are listed here.

Step 1. At \mathbf{x}^0 calculate $f(\mathbf{x}^0)$. Let

$$\mathbf{s}^0 = -\nabla f(\mathbf{x}^0)$$

Step 2. Save $\nabla f(\mathbf{x}^0)$ and compute

$$\mathbf{x}^1 = \mathbf{x}^0 + \alpha^0 \mathbf{s}^0$$

by minimizing $f(\mathbf{x})$ with respect to α in the \mathbf{s}^0 direction (i.e., carry out a unidimensional search for α^0).

Step 3. Calculate $f(\mathbf{x}^1)$, $\nabla f(\mathbf{x}^1)$. The new search direction is a linear combination of \mathbf{s}^0 and $\nabla f(\mathbf{x}^1)$:

$$\mathbf{s}^1 = -\nabla f(\mathbf{x}^1) + \mathbf{s}^0 \frac{\nabla^T f(\mathbf{x}^1) \nabla f(\mathbf{x}^1)}{\nabla^T f(\mathbf{x}^0) \nabla f(\mathbf{x}^0)}$$

For the k th iteration the relation is

$$\mathbf{s}^{k+1} = -\nabla f(\mathbf{x}^{k+1}) + \mathbf{s}^k \frac{\nabla^T f(\mathbf{x}^{k+1}) \nabla f(\mathbf{x}^{k+1})}{\nabla^T f(\mathbf{x}^k) \nabla f(\mathbf{x}^k)} \quad (6.6)$$

For a quadratic function it can be shown that these successive search directions are conjugate. After n iterations ($k = n$), the quadratic function is minimized. For a nonquadratic function, the procedure cycles again with \mathbf{x}^{n+1} becoming \mathbf{x}^0 .

Step 4. Test for convergence to the minimum of $f(\mathbf{x})$. If convergence is not attained, return to step 3.

Step n . Terminate the algorithm when $\|\nabla f(\mathbf{x}^k)\|$ is less than some prescribed tolerance.

Note that if the ratio of the inner products of the gradients from stage $k + 1$ relative to stage k is very small, the conjugate gradient method behaves much like the steepest descent method. One difficulty is the linear dependence of search directions, which can be resolved by periodically restarting the conjugate gradient method with a steepest descent search (step 1). The proof that Equation (6.6) yields conjugate directions and quadratic convergence was given by Fletcher and Reeves (1964).

In doing the line search we can minimize a quadratic approximation in a given search direction. This means that to compute the value for α for the relation $\mathbf{x}^{k+1} = \mathbf{x}^k + \alpha \mathbf{s}^k$ we must minimize

$$f(\mathbf{x}) = f(\mathbf{x}^k + \alpha \mathbf{s}^k) = f(\mathbf{x}^k) + \nabla^T f(\mathbf{x}^k) \alpha \mathbf{s}^k + \frac{1}{2} (\alpha \mathbf{s}^k)^T \mathbf{H}(\mathbf{x}^k) (\alpha \mathbf{s}^k) \quad (6.7)$$

where $\Delta \mathbf{x}^k = \alpha \mathbf{s}^k$. To get the minimum of $f(\mathbf{x}^k + \alpha \mathbf{s}^k)$, we differentiate Equation (6.7) with respect to α and equate the derivative to zero

$$\frac{df(\mathbf{x}^k + \alpha \mathbf{s}^k)}{d\alpha} = 0 = \nabla^T f(\mathbf{x}^k) \mathbf{s}^k + (\mathbf{s}^k)^T \mathbf{H}(\mathbf{x}^k) \alpha \mathbf{s}^k \quad (6.8)$$

with the result

$$\alpha^{\text{opt}} = -\frac{\nabla^T f(\mathbf{x}^k) \mathbf{s}^k}{(\mathbf{s}^k)^T \mathbf{H}(\mathbf{x}^k) \mathbf{s}^k} \quad (6.9)$$

For additional details concerning the application of conjugate gradient methods, especially to large-scale and sparse problems, refer to Fletcher (1980), Gill et al. (1981), Dembo et al. (1982), and Nash and Sofer (1996).

EXAMPLE 6.2 APPLICATION OF THE FLETCHER-REEVES CONJUGATE GRADIENT ALGORITHM

We solve the problem known as Rosenbrock's function

$$\text{Minimize: } f(\mathbf{x}) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$$

starting at $\mathbf{x}^{(0)} = [-1.2 \ 1.0]^T$. The first few stages of the Fletcher-Reeves procedure are listed in Table E6.2. The trajectory as it moves toward the optimum is shown in Figure E6.2.

TABLE E6.2
Results for Example 6.2 using the Fletcher-Reeves method

Iteration	Number of function calls	$f(\mathbf{x})$	x_1	x_2	$\frac{\partial f(\mathbf{x})}{\partial x_1}$	$\frac{\partial f(\mathbf{x})}{\partial x_2}$
0	1	24.2	-1.2	1.0	-215.6	-88.00
1	4	4.377945	-1.050203	1.061141	-21.65	-8.357
5	14	3.165142	-0.777190	0.612232	-1.002	-1.6415
10	28	1.247687	-0.079213	-0.025322	-3.071	-5.761
15	41	0.556612	0.254058	0.063189	-1.354	-0.271
20	57	0.147607	0.647165	0.403619	3.230	-3.040
25	69	0.024667	0.843083	0.710119	-0.0881	-0.1339
30	80	0.0000628	0.995000	0.989410	0.2348	-0.1230
35	90	1.617×10^{-15}	1.000000	1.000000	-1.60×10^{-8}	-3.12×10^{-8}

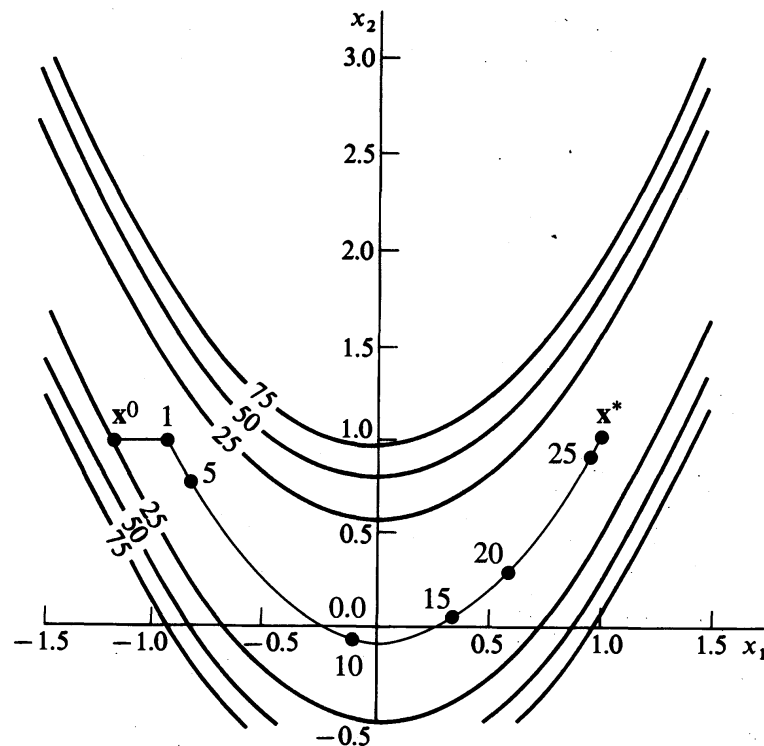


FIGURE E6.2

Search trajectory for the Fletcher-Reeves algorithm (the numbers designate the iteration).

6.3 NEWTON'S METHOD

From one viewpoint the search direction of steepest descent can be interpreted as being orthogonal to a linear approximation (tangent to) of the objective function at point \mathbf{x}^k ; examine Figure 6.9a. Now suppose we make a quadratic approximation of $f(\mathbf{x})$ at \mathbf{x}^k

$$f(\mathbf{x}) \approx f(\mathbf{x}^k) + \nabla^T f(\mathbf{x}^k) \Delta \mathbf{x}^k + \frac{1}{2} (\Delta \mathbf{x}^k)^T \mathbf{H}(\mathbf{x}^k) \Delta \mathbf{x}^k \quad (6.10)$$

where $\mathbf{H}(\mathbf{x}^k)$ is the Hessian matrix of $f(\mathbf{x})$ defined in Chapter 4 (the matrix of second partial derivatives with respect to \mathbf{x} evaluated at \mathbf{x}^k). Then it is possible to take into account the curvature of $f(\mathbf{x})$ at \mathbf{x}^k in determining a search direction as described later on.

Newton's method makes use of the second-order (quadratic) approximation of $f(\mathbf{x})$ at \mathbf{x}^k and thus employs second-order information about $f(\mathbf{x})$, that is, information obtained from the second partial derivatives of $f(\mathbf{x})$ with respect to the independent variables. Thus, it is possible to take into account the curvature of $f(\mathbf{x})$ at \mathbf{x}^k and identify better search directions than can be obtained via the gradient method. Examine Figure 6.9b.

The minimum of the quadratic approximation of $f(\mathbf{x})$ in Equation (6.10) is obtained by differentiating (6.10) with respect to each of the components of $\Delta \mathbf{x}$ and equating the resulting expressions to zero to give

$$\nabla f(\mathbf{x}) = \nabla f(\mathbf{x}^k) + \mathbf{H}(\mathbf{x}^k) \Delta \mathbf{x}^k = 0 \quad (6.11)$$

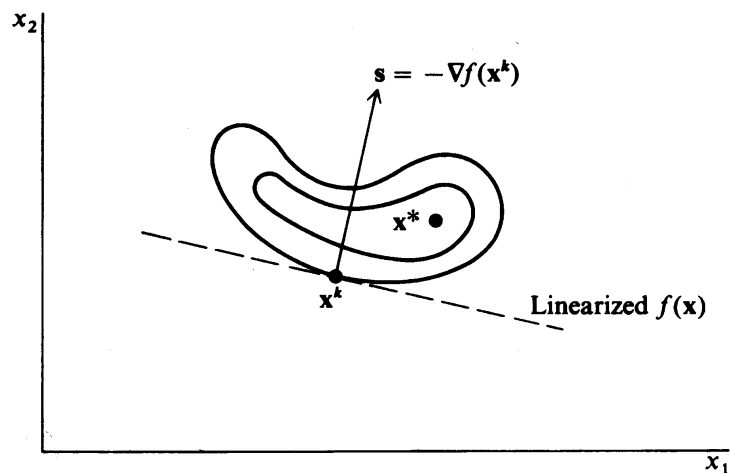
or

$$\mathbf{x}^{k+1} - \mathbf{x}^k = \Delta \mathbf{x}^k = - [\mathbf{H}(\mathbf{x}^k)]^{-1} \nabla f(\mathbf{x}^k) \quad (6.12)$$

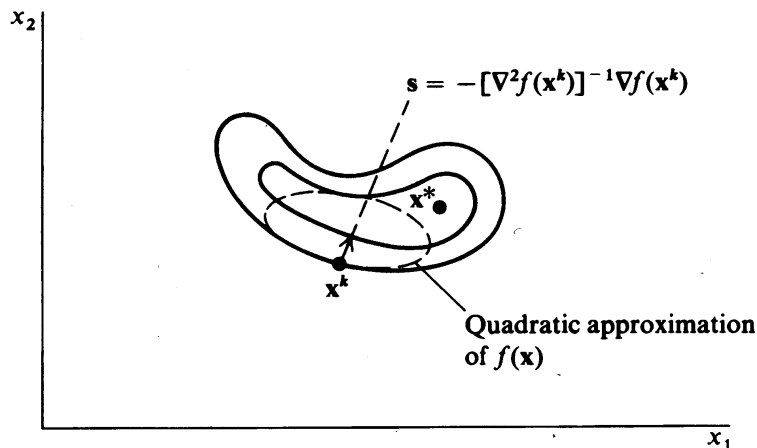
where $[\mathbf{H}(\mathbf{x}^k)]^{-1}$ is the inverse of the Hessian matrix $\mathbf{H}(\mathbf{x}^k)$. Equation (6.12) reduces to Equation (5.5) for a one-dimensional search.

Note that both the direction *and* step length are specified as a result of Equation (6.11). If $f(\mathbf{x})$ is actually quadratic, only one step is required to reach the minimum of $f(\mathbf{x})$. For a general nonlinear objective function, however, the minimum of $f(\mathbf{x})$ cannot be reached in one step, so that Equation (6.12) can be modified to conform to Equation (6.7) by introducing the parameter for the step length into (6.12).

$$\mathbf{x}^{k+1} - \mathbf{x}^k = -\alpha^k [\mathbf{H}(\mathbf{x}^k)]^{-1} \nabla f(\mathbf{x}^k) \quad (6.13)$$



(a) Steepest descent: first-order approximation (linearization) of $f(\mathbf{x})$ at \mathbf{x}^k



(b) Newton's method: second-order (quadratic) approximation of $f(\mathbf{x})$ at \mathbf{x}^k

FIGURE 6.9

Comparison of steepest descent with Newton's method from the viewpoint of objective function approximation.

Observe that the search direction \mathbf{s} is now given (for minimization) by

$$\mathbf{s}^k = -[\mathbf{H}(\mathbf{x}^k)]^{-1} \nabla f(\mathbf{x}^k) \quad (6.14)$$

and that the step length is α^k . The step length α^k can be evaluated numerically as described in Chapter 5. Equation (6.13) is applied iteratively until some termination criteria are satisfied. For the "pure" version of Newton's method, $\alpha = 1$ on each step. However, this version often does not converge if the initial point is not close enough to a local minimum.

Also note that to evaluate $\Delta \mathbf{x}$ in Equation (6.12), a matrix inversion is not necessarily required. You can take its precursor, Equation (6.11), and solve the following set of linear equations for $\Delta \mathbf{x}^k$

$$\mathbf{H}(\mathbf{x}^k) \Delta \mathbf{x}^k = -\nabla f(\mathbf{x}^k) \quad (6.15)$$

a procedure that often leads to less round-off error than calculating \mathbf{s} via the inversion of a matrix.

EXAMPLE 6.3 APPLICATION OF NEWTON'S METHOD TO A CONVEX QUADRATIC FUNCTION

We minimize the function

$$f(\mathbf{x}) = 4x_1^2 + x_2^2 - 2x_1x_2$$

starting at $\mathbf{x}^0 = [1 \ 1]^T$

$$\nabla f(\mathbf{x}) = \begin{bmatrix} 8x_1 - 2x_2 \\ 2x_2 - 2x_1 \end{bmatrix}$$

$$\mathbf{H}(\mathbf{x}) = \begin{bmatrix} 8 & -2 \\ -2 & 2 \end{bmatrix} \quad \mathbf{H}^{-1}(\mathbf{x}) = \begin{bmatrix} \frac{1}{6} & \frac{1}{6} \\ \frac{1}{6} & \frac{2}{3} \end{bmatrix}$$

with $\alpha = 1$,

$$\Delta \mathbf{x}^0 = -\mathbf{H}^{-1} \nabla f(\mathbf{x}^0) = -\begin{bmatrix} \frac{1}{6} & \frac{1}{6} \\ \frac{1}{6} & \frac{2}{3} \end{bmatrix} \begin{bmatrix} 6 \\ 0 \end{bmatrix} = \begin{bmatrix} -1 \\ -1 \end{bmatrix}$$

hence,

$$\mathbf{x}^1 = \mathbf{x}^* = \mathbf{x}^0 + \Delta \mathbf{x}^0 = \begin{bmatrix} 1 \\ 1 \end{bmatrix} + \begin{bmatrix} -1 \\ -1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$f(\mathbf{x}^*) = 0$$

Instead of taking the inverse of \mathbf{H} , we can solve Equation (6.15)

$$\begin{bmatrix} 8 & -2 \\ -2 & 2 \end{bmatrix} \begin{bmatrix} \Delta x_1^0 \\ \Delta x_2^0 \end{bmatrix} = -\begin{bmatrix} 6 \\ 0 \end{bmatrix}$$

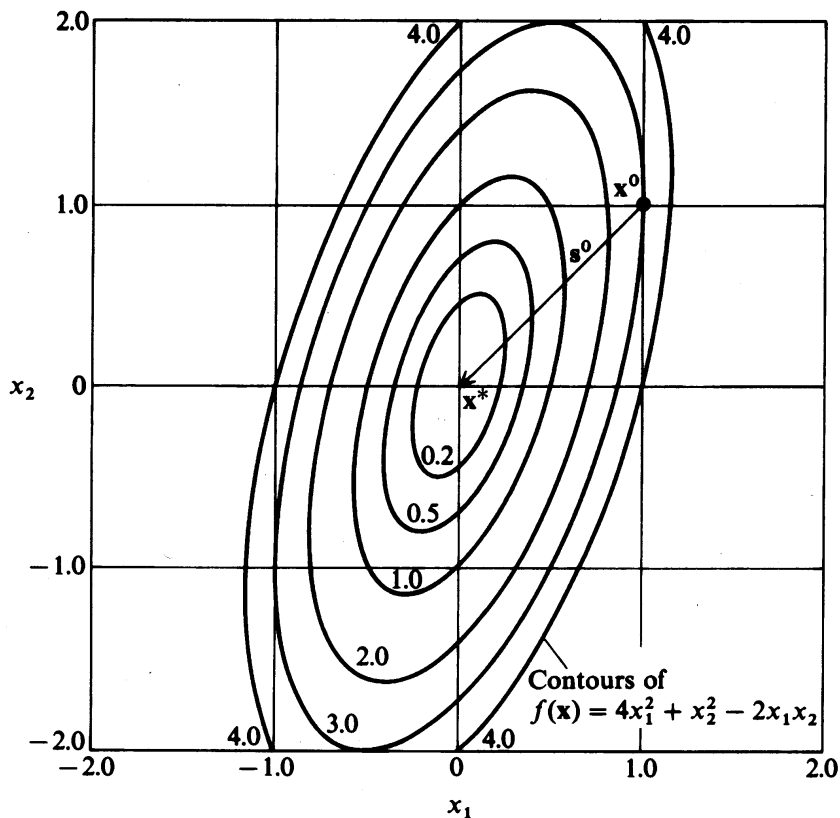


FIGURE E6.3

which gives

$$\Delta x_1^0 = -1$$

$$\Delta x_2^0 = -1$$

as before. The search direction $\mathbf{s}^0 = -\mathbf{H}^{-1} \nabla f(\mathbf{x}^0)$ is shown in Figure E6.3

EXAMPLE 6.4 APPLICATION OF NEWTON'S METHOD AND QUADRATIC CONVERGENCE

If we minimize the nonquadratic function

$$f(\mathbf{x}) = (x_1 - 2)^4 + (x_1 - 2)^2 x_2^2 + (x_2 + 1)^2$$

from the starting point of (1, 1), can you show that Newton's method exhibits quadratic convergence? *Hint:* Show that

$$\frac{\|\mathbf{x}^{k+1} - \mathbf{x}^*\|}{\|\mathbf{x}^k - \mathbf{x}^*\|^2} < c \quad (\text{see Section 5.3})$$

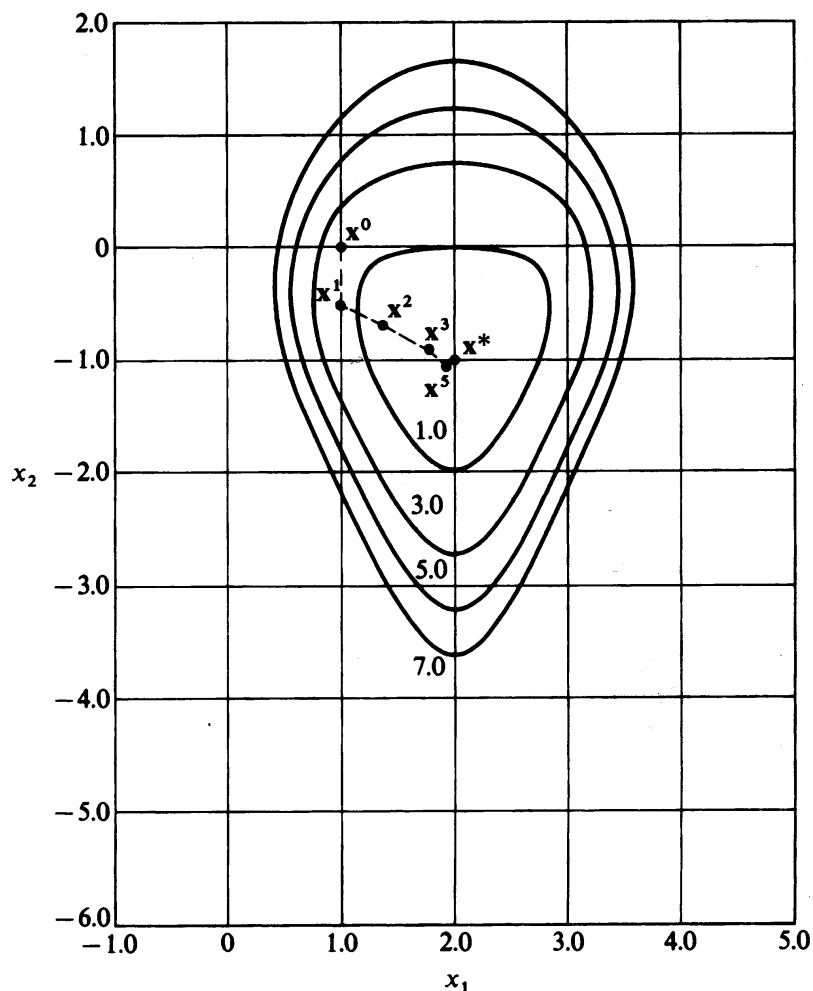


FIGURE E6.4

Solution. Newton's method produces the following sequences of values for x_1 , x_2 , and $[f(\mathbf{x}^{k+1}) - f(\mathbf{x}^k)]$ (you should try to verify the calculations shown in the following table; the trajectory is traced in Figure E6.4).

Iteration	x_1	x_2	$f(\mathbf{x}^{k+1}) - f(\mathbf{x}^k)$
0	1.000000	1.000000	6.000
1	1.000000	-0.500000	1.500
2	1.391304	-0.695652	4.09×10^{-1}
3	1.745944	-0.948798	6.49×10^{-2}
4	1.986278	-1.048208	2.53×10^{-3}
5	1.998734	-1.000170	1.63×10^{-6}
6	1.999996	-1.000002	2.75×10^{-12}

You can calculate between iterations 2 and 3 that $c = 0.55$; and between 3 and 4 that $c \approx 0.74$. Hence, quadratic convergence can be demonstrated numerically.

Newton's method usually requires the fewest iterations of all the methods discussed in this chapter, but it has the following disadvantages:

1. The method does not necessarily find the global solution if multiple local solutions exist, but this is a characteristic of all the methods described in this chapter.
2. It requires the solution of a set of n symmetric linear equations.
3. It requires both first and second partial derivatives, which may not be practical to obtain.
4. Using a step size of unity, the method may not converge.

Difficulty 3 can be ameliorated by using (properly) finite difference approximation as substitutes for derivatives. To overcome difficulty 4, two classes of methods exist to modify the "pure" Newton's method so that it is guaranteed to converge to a local minimum from an arbitrary starting point. The first of these, called *trust region methods*, minimize the quadratic approximation, Equation (6.10), within an elliptical region, whose size is adjusted so that the objective improves at each iteration; see Section 6.3.2. The second class, line search methods, modifies the pure Newton's method in two ways: (1) instead of taking a step size of one, a line search is used and (2) if the Hessian matrix $\mathbf{H}(\mathbf{x}^k)$ is not positive-definite, it is replaced by a positive-definite matrix that is "close" to $\mathbf{H}(\mathbf{x}^k)$. This is motivated by the easily verified fact that, if $\mathbf{H}(\mathbf{x}^k)$ is positive-definite, the Newton direction

$$\mathbf{s}^k = -[\mathbf{H}(\mathbf{x}^k)]^{-1} \nabla f(\mathbf{x}^k)$$

is a descent direction, that is

$$\nabla f^T(\mathbf{x}^k) \mathbf{s}^k < 0$$

If $f(\mathbf{x})$ is convex, $\mathbf{H}(\mathbf{x})$ is positive-semidefinite at all points \mathbf{x} and is usually positive-definite. Hence Newton's method, using a line search, converges. If $f(\mathbf{x})$ is not strictly convex (as is often the case in regions far from the optimum), $\mathbf{H}(\mathbf{x})$ may not be positive-definite everywhere, so one approach to forcing convergence is to replace $\mathbf{H}(\mathbf{x})$ by another positive-definite matrix. The Marquardt-Levenberg method is one way of doing this, as discussed in the next section.

6.3.1 Forcing the Hessian Matrix to Be Positive-Definite

Marquardt (1963), Levenberg (1944), and others have suggested that the Hessian matrix of $f(\mathbf{x})$ be modified on each stage of the search as needed to ensure that the modified $\mathbf{H}(\mathbf{x})$, $\tilde{\mathbf{H}}(\mathbf{x})$, is positive-definite and well conditioned. The procedure adds elements to the diagonal elements of $\mathbf{H}(\mathbf{x})$

$$\tilde{\mathbf{H}}(\mathbf{x}) = [\mathbf{H}(\mathbf{x}) + \beta \mathbf{I}] \quad (6.16)$$

where β is a positive constant large enough to make $\tilde{\mathbf{H}}(\mathbf{x})$ positive-definite when $\mathbf{H}(\mathbf{x})$ is not. Note that with a β sufficiently large, $\beta \mathbf{I}$ can overwhelm $\mathbf{H}(\mathbf{x})$ and the minimization approaches a steepest descent search.

TABLE 6.1
A modified Marquardt method

Step 1

Pick \mathbf{x}^0 the starting point. Let $\varepsilon =$ convergence criterion.

Step 2

Set $k = 0$. Let $\beta^0 = 10^3$.

Step 3

Calculate $\nabla f(\mathbf{x}^k)$.

Step 4

Is $\|\nabla f(\mathbf{x}^k)\| < \varepsilon$? If yes, terminate. If no, continue.

Step 5

Solve $(\mathbf{H}(\mathbf{x}^k) + \beta^k \mathbf{I}) \mathbf{s}^k = -\nabla f(\mathbf{x}^k)$ for \mathbf{s}^k .

Step 6

If $\nabla f^T(\mathbf{x}^k) \mathbf{s}^k < 0$, go to step 8.

Step 7

Set $\beta^k = 2\beta^k$ and go to step 5.

Step 8

Choose α^k by a line search procedure so that

$$f(\mathbf{x}^k + \alpha^k \mathbf{s}^k) < f(\mathbf{x}^k)$$

Step 9

If certain conditions are met (Dennis and Schnabel, 1996), reduce β .

Go to step 3 with k replaced by $k + 1$.

A simpler procedure that may result in a suitable value of β is to apply a modified Cholesky factorization as follows:

$$\mathbf{H}(\mathbf{x}^k) + \mathbf{D} = \mathbf{L}\mathbf{L}^T \quad (6.17)$$

where \mathbf{D} is a diagonal matrix with nonnegative elements [$d_{ii} = 0$ if $\mathbf{H}(\mathbf{x}^k)$ is positive-definite] and \mathbf{L} is a lower triangular matrix. Upper bounds on the elements in \mathbf{D} are calculated using the Gershgorin circle theorem [see Dennis and Schnabel (1996) for details].

A simple algorithm based on an arbitrary adjustment of β (a modified Marquardt's method) is listed in Table 6.1.

EXAMPLE 6.5 APPLICATION OF MARQUARDT'S METHOD

The algorithm listed in Table 6.1 is to be applied to Rosenbrock's function $f(\mathbf{x}) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$ starting at $\mathbf{x}^0 = [-1.2 \quad 1.0]^T$ with $\mathbf{H}^0 = \mathbf{H}(\mathbf{x}^0)$.

TABLE E6.5
Marquardt's method

$f(\mathbf{x})$	x_1	x_2	$\frac{\partial f(\mathbf{x})}{\partial x_1}$	$\frac{\partial f(\mathbf{x})}{\partial x_2}$	Elements of $[\mathbf{H}(\mathbf{x}^k) + \beta \mathbf{I}]^{-1}$			
					\tilde{h}_{11}^{-1}	\tilde{h}_{12}^{-1}	\tilde{h}_{21}^{-1}	\tilde{h}_{22}^{-1}
24.2000	-1.2000	1.0000	-215.6000	-88.0000	0.0005	-0.0002	-0.0002	0.0009
4.1498	-1.0315	1.0791	2.1844	3.0284	0.0005	-0.0002	-0.0002	0.0009
4.1173	-1.0289	1.0557	-5.2448	-0.5768	0.0014	-0.0013	-0.0013	0.0034
3.9642	-0.9412	0.9301	12.7861	8.8552	0.0037	-0.0059	-0.0059	0.0130
3.4776	-0.8542	0.7098	-10.5031	-3.9772	0.0195	-0.0341	-0.0341	0.0641
2.7527	-0.6028	0.3206	-13.5391	-8.5706	0.0399	-0.0669	-0.0669	0.1170
1.9132	-0.3167	0.0580	-7.9993	-8.4706	0.0464	-0.0557	-0.0557	0.0718
1.1890	-0.0313	-0.0344	-2.5059	-7.0832	0.0519	-0.0328	-0.0328	0.0258
0.6885	0.2278	0.0215	1.2242	-6.0759	0.0616	-0.0039	-0.0039	0.0052
0.3266	0.4570	0.2031	3.2402	-4.5160	0.0706	0.0322	0.0322	0.0196
0.1275	0.6846	0.4520	3.9595	3.3523	0.0906	0.0861	0.0861	0.0868
0.0237	0.8705	0.7495	2.6299	-1.6593	0.1148	0.1573	0.1573	0.2203
0.0006	0.9870	0.9721	0.7700	-0.4033	0.1880	0.3273	0.3273	0.5748
0.0000	0.9974	0.9949	-0.0589	0.0269	0.3563	0.7033	0.7033	1.3932
0.0000	0.9999	0.9999	-0.0004	0.0002	0.5138	1.0249	1.0249	2.0494
0.0000	1.0000	1.0000	0.0000	-0.0000	0.5001	1.0001	1.0001	2.0050
0.0000	1.0000	1.0000						

A quadratic interpolation subroutine was used to minimize in each search direction. Table E6.5 lists the values of $f(\mathbf{x})$, \mathbf{x} , $\nabla f(\mathbf{x})$, and the elements of $[\mathbf{H}(\mathbf{x}) + \beta \mathbf{I}]^{-1}$ for each stage of the minimization. A total of 96 function evaluations and 16 calls to the gradient evaluation subroutine were needed.

6.3.2 Movement in the Search Direction

Up to this point we focused on calculating \mathbf{H} or \mathbf{H}^{-1} , from which the search direction \mathbf{s} can be ascertained via Equation (6.14) or $\Delta \mathbf{x}$ from Equation (6.15) (for minimization). In this section we discuss briefly how far to proceed in the search direction, that is, select a step length, for a general function $f(\mathbf{x})$. If $\Delta \mathbf{x}$ is calculated from Equations (6.12) or (6.15), $\alpha = 1$ and the step is a Newton step. If $\alpha \neq 1$, then any procedure can be used to calculate α as discussed in Chapter 5.

Line search. The oldest and simplest method of calculating α to obtain $\Delta \mathbf{x}$ is via a *unidimensional line search*. In a given direction that reduces $f(\mathbf{x})$, take a step, or a sequence of steps yielding an overall step, that reduces $f(\mathbf{x})$ to some acceptable degree. This operation can be carried out by any of the one-dimensional search

techniques described in Chapter 5. Early investigators always minimized $f(\mathbf{x})$ as accurately as possible in a search direction \mathbf{s} , but subsequent experience, and to some extent theoretical results, have indicated that such a concept is invalid. Good algorithms first calculate a full Newton step ($\alpha = 1$) to get \mathbf{x}^{k+1} , and if $f(\mathbf{x}^k)$ is not reduced, backtrack in some systematic way toward \mathbf{x}^k . Failure to take the full Newton step in the first iteration leads to loss of the advantages of Newton's method near the minimum, where convergence is slow. To avoid very small decreases in $f(\mathbf{x})$, most algorithms require that the average rate of descent from \mathbf{x}^k to \mathbf{x}^{k+1} be at least some prescribed fraction of the initial rate of descent in the search direction. Mathematically this means (Armijo, 1966)

$$f(\mathbf{x}^k + \alpha \mathbf{s}^k) \leq f(\mathbf{x}^k) + \gamma \alpha \nabla^T f(\mathbf{x}^k) \mathbf{s}^k \quad (6.18)$$

Examine Figure 6.10. In practice γ is often chosen to be very small, about 10^{-4} , so just a small decrease in the function value is required.

Backtracking can be accomplished in any of the ways outlined in Chapter 5 but with the objective of locating an \mathbf{x}^{k+1} for which $f(\mathbf{x}^{k+1}) < f(\mathbf{x}^k)$ but moving as far as possible in the direction \mathbf{s}^k from \mathbf{x}^k . The minimum of $f(\mathbf{x}^k + \alpha \mathbf{s}^k)$ does not have to be found exactly. As an example of one procedure, at \mathbf{x}^k , where $\alpha = 0$, you know two pieces of information about $f(\mathbf{x}^k + \alpha \mathbf{s}^k)$: the values of $f(\mathbf{x}^k)$ and $\nabla^T f(\mathbf{x}^k) \mathbf{s}^k$. After the Newton step ($\alpha = 1$) you know the value of $f(\mathbf{x}^k + \mathbf{s}^k)$. From these three pieces of information you can make a quadratic interpolation to get the value α where the objective function $f(\alpha)$ has a minimum:

$$\hat{\alpha} = -\frac{\nabla^T f(\mathbf{x}^k) \mathbf{s}^k}{2[f(\mathbf{x}^k + \mathbf{s}^k) - f(\mathbf{x}^k) - \nabla^T f(\mathbf{x}^k) \mathbf{s}^k]} \quad (6.19)$$

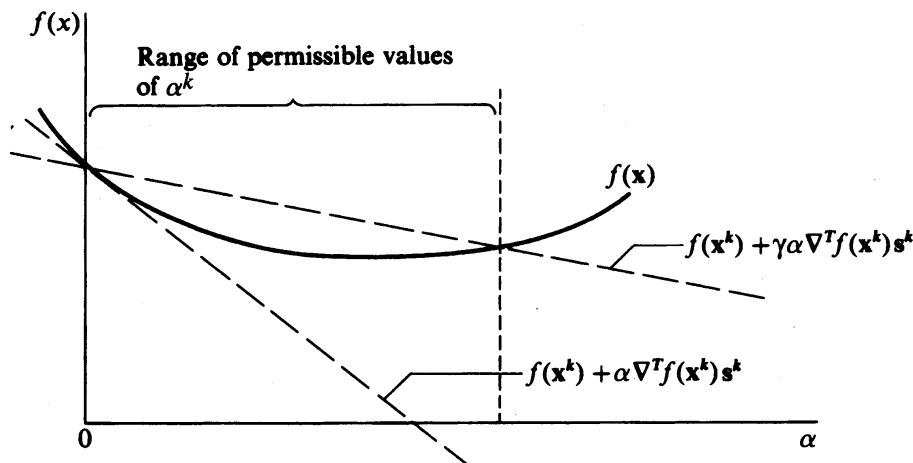


FIGURE 6.10

Range of acceptable values for choice of α^k to meet criterion (6.20) with $\gamma = 0.02$.

After $\hat{\alpha}$ is obtained, if additional backtracking is needed, cubic interpolation can be carried out. We suggest that if $\hat{\alpha}$ is too small, say $\hat{\alpha} < 0.1$, try $\hat{\alpha} = 0.1$ instead.

Trust regions. The name *trust region* refers to the region in which the quadratic model can be “trusted” to represent $f(\mathbf{x})$ reasonably well. In the unidimensional line search, the search direction is retained but the step length is reduced if the Newton step proves to be unsatisfactory. In the trust region approach, a shorter step length is selected and *then* the search direction determined. Refer to Dennis and Schnabel (1996) and Section 8.5.1 for details.

The trust region approach estimates the length of a maximal successful step from \mathbf{x}^k . In other words, $\|\mathbf{x}\| < \rho$, the bound on the step. Figure 6.11 shows $f(\mathbf{x})$, the quadratic model of $f(\mathbf{x})$, and the desired trust region. First, an initial estimate of ρ or the step bound has to be determined. If knowledge about the problem does

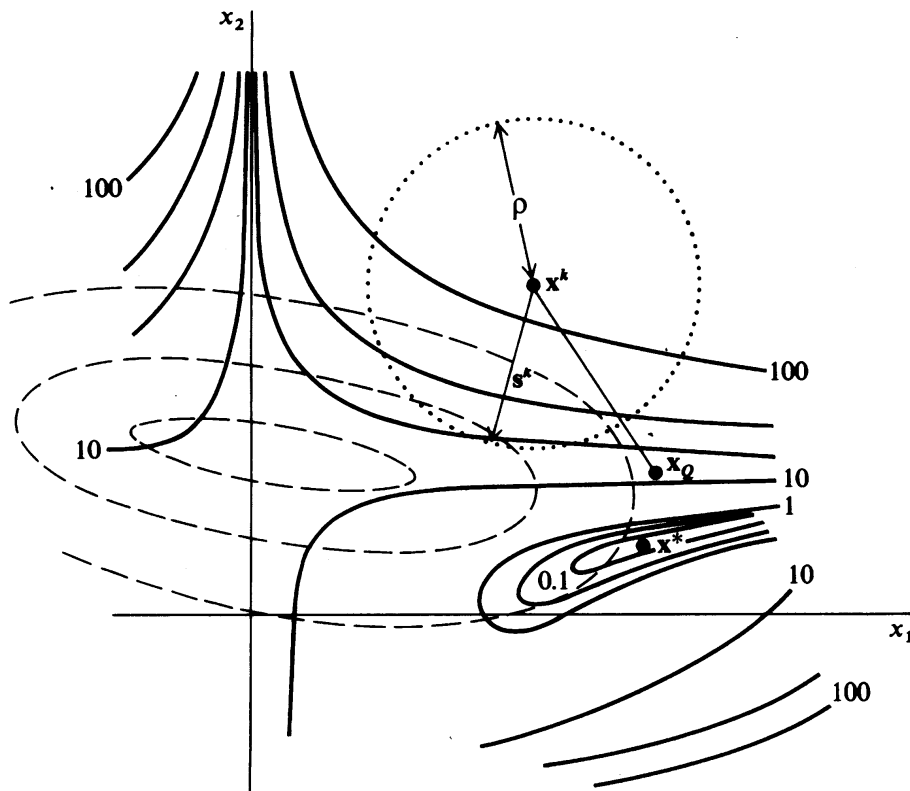


FIGURE 6.11

Representation of the trust region to select the step length. Solid lines are contours of $f(\mathbf{x})$. Dashed lines are contours of the convex quadratic approximation of $f(\mathbf{x})$ at \mathbf{x}^k . The dotted circle is the trust region boundary in which δ is the step length. \mathbf{x}_0 is the minimum of the quadratic model for which $\hat{\mathbf{H}}(\mathbf{x})$ is positive-definite.

not help, Powell (1970) suggested using the distance to the minimizer of the quadratic model of $f(\mathbf{x})$ in the direction of steepest descent from \mathbf{x}^k , the so-called Cauchy point. Next, some curve or piecewise linear function is determined with an initial direction of steepest descent so that the tentative point \mathbf{x}^{k+1} lies on the curve and is less than ρ . Figure 6.11 shows \mathbf{s} as a straight line of one segment. The trust region is updated, and the sequence is continued. Heuristic parameters are usually required, such as minimum and maximum step lengths, scaling \mathbf{s} , and so forth.

6.3.3 Termination

No single stopping criterion will suffice for Newton's method or any of the optimization methods described in this chapter. The following simultaneous criteria are recommended to avoid scaling problems:

$$|f(\mathbf{x}^{k+1}) - f(\mathbf{x}^k)| < \varepsilon_1(1 + |f(\mathbf{x}^k)|) \quad (6.20)$$

where the "one" on the right-hand side is present to ensure that the right-hand side is not too small when $f(\mathbf{x}^k)$ approaches zero. Also

$$\|\mathbf{x}^{k+1} - \mathbf{x}^k\| < \varepsilon_2(1 + \|\mathbf{x}^k\|) \quad (6.21)$$

and

$$\|\nabla f(\mathbf{x}^k)\| < \varepsilon_3 \quad (6.22)$$

6.3.4 Safeguarded Newton's Method

Several numerical subroutine libraries contain "safeguarded" Newton codes using the ideas previously discussed. When first and second derivatives can be computed quickly and accurately, a good safeguarded Newton code is fast, reliable, and locates a local optimum very accurately. We discuss this NLP software in Section 8.9.

6.3.5 Computation of Derivatives

From numerous tests involving optimization of nonlinear functions, methods that use derivatives have been demonstrated to be more efficient than those that do not. By replacing analytical derivatives with their finite difference substitutes, you can avoid having to code formulas for derivatives. Procedures that use second-order information are more accurate and require fewer iterations than those that use only first-order information (gradients), but keep in mind that usually the second-order information may be only approximate as it is based not on second derivatives themselves but their finite difference approximations.

6.4 QUASI-NEWTON METHODS

Procedures that compute a search direction using only first derivatives of f provide an attractive alternative to Newton's method. The most popular of these are the quasi-Newton methods that replace $\mathbf{H}(\mathbf{x}^k)$ in Equation (6.11) by a positive-definite approximation $\tilde{\mathbf{H}}^k$:

$$\tilde{\mathbf{H}}^k \mathbf{s}^k = -\nabla f(\mathbf{x}^k) \quad (6.23)$$

$\tilde{\mathbf{H}}^k$ is initialized as any positive-definite symmetric matrix (often the identity matrix or a diagonal matrix) and is updated after each line search using the changes in \mathbf{x} and in $\nabla f(\mathbf{x})$ over the last two points, as measured by the vectors

$$\mathbf{d}^k = \mathbf{x}^{k+1} - \mathbf{x}^k \quad (6.24)$$

and

$$\mathbf{y}^k = \nabla f(\mathbf{x}^{k+1}) - \nabla f(\mathbf{x}^k) \quad (6.25)$$

One of the most efficient and widely used updating formula is the BFGS update. Broyden (1970), Fletcher (1970), Goldfarb (1970), and Shanno (1970) independently published this algorithm in the same year, hence the combined name BFGS. Here the approximate Hessian is given by

$$\tilde{\mathbf{H}}^{k+1} = \tilde{\mathbf{H}}^k + \frac{\mathbf{y}^k (\mathbf{y}^k)^T}{(\mathbf{d}^k)^T \mathbf{y}^k} - \frac{(\tilde{\mathbf{H}}^k \mathbf{d}^k) (\tilde{\mathbf{H}}^k \mathbf{d}^k)^T}{(\mathbf{d}^k)^T \tilde{\mathbf{H}}^k \mathbf{d}^k} \quad (6.26)$$

If $\tilde{\mathbf{H}}^k$ is positive-definite and $(\mathbf{d}^k)^T \mathbf{y}^k > 0$, it can be shown that $\tilde{\mathbf{H}}^{k+1}$ is positive-definite (Dennis and Schnabel, 1996, Chapter 9). The condition $(\mathbf{d}^k)^T \mathbf{y}^k > 0$ can be interpreted geometrically, since

$$\begin{aligned} (\mathbf{d}^k)^T \mathbf{y}^k &= \alpha^k (\mathbf{s}^k)^T [\nabla f(\mathbf{x}^{k+1}) - \nabla f(\mathbf{x}^k)] \\ &= \alpha^k [(\mathbf{s}^k)^T \nabla f(\mathbf{x}^{k+1}) - (\mathbf{s}^k)^T \nabla f(\mathbf{x}^k)] \\ &= \alpha^k (\text{slope2} - \text{slope1}) \end{aligned}$$

The quantity slope2 is the slope of the line search objective function $g^k(\alpha)$ at $\alpha = \alpha^k$ (see Figure 6.8) and slope1 is its slope at $\alpha = 0$, so $(\mathbf{d}^k)^T \mathbf{y}^k > 0$ if and only if slope2 > slope1. This condition is always satisfied if f is strictly convex. A good line search routine attempts to meet this condition; if it is not met, then $\tilde{\mathbf{H}}^k$ is not updated.

If the BFGS algorithm is applied to a positive-definite quadratic function of n variables and the line search is exact, it will minimize the function in at most n iterations (Dennis and Schnabel, 1996, Chapter 9). This is also true for some other updating formulas. For nonquadratic functions, a good BFGS code usually requires more iterations than a comparable Newton implementation and may not be as accurate. Each BFGS iteration is generally faster, however, because second derivatives are not required and the system of linear equations (6.15) need not be solved.

EXAMPLE 6.6 APPLICATION OF THE BFGS METHOD

Apply the BFGS method to find the minimum of the function $f(\mathbf{x}) = x_1^4 - 2x_2x_1^2 + x_2^2 + x_1^2 - 2x_1 + 5$.

Use a starting point of (1,2) and terminate the search when f changes less than 0.00005 between iterations. The contour plot for the function was shown in Figure 5.7.

Solution. Using the Optimization Toolbox from MATLAB, the BFGS method requires 20 iterations before the search is terminated, as shown below.

TABLE E6.6
BFGS method

Iteration	x_1	x_2	$f(x_1, x_2)$	$\frac{\partial f}{\partial x_1}$	$\frac{\partial f}{\partial x_2}$
	1.00000	2.00000	5.00000	-4.00000	2.00000
1	1.29611	1.82473	4.10866	-0.15866	0.28966
2	1.29192	1.73556	4.08964	0.24022	0.13299
3	1.22980	1.63069	4.06680	-0.12218	0.23654
4	1.22409	1.54972	4.05285	0.19694	0.10263
5	1.17160	1.46528	4.03803	-0.09085	0.18524
6	1.16530	1.39587	4.02876	0.15372	0.07589
7	1.12318	1.33087	4.01998	-0.06513	0.13867
8	1.11718	1.27501	4.01446	0.11408	0.05383
9	1.08519	1.22728	4.00972	-0.04507	0.09927
10	1.08012	1.18504	4.00676	0.08077	0.03678
11	1.05705	1.15150	4.00442	-0.03024	0.06828
12	1.05314	1.12129	4.00297	0.05494	0.02438
13	1.03725	1.09861	4.00190	-0.01977	0.04544
14	1.03444	1.07795	4.00125	0.03623	0.01578
15	1.02386	1.06305	4.00079	-0.01269	0.02950
16	1.02195	1.04940	4.00051	0.02335	0.01005
17	1.01509	1.03981	4.00032	-0.00803	0.01882
18	1.01382	1.03100	4.00020	0.01482	0.00632
19	1.00945	1.02492	4.00012	-0.00503	0.01186
20	1.00863	1.01932	4.00008	0.00930	0.00395

For problems with hundreds or thousands of variables, storing and manipulating the matrices $\tilde{\mathbf{H}}^k$ or $\nabla^2 f(\mathbf{x}^k)$ requires much time and computer memory, making conjugate gradient methods more attractive. These compute \mathbf{s}^k using formulas involving no matrices. The Fletcher–Reeves method uses

$$\mathbf{s}^0 = -\nabla f(\mathbf{x}^0)$$

$$\mathbf{s}^k = -\nabla f(\mathbf{x}^k) + \beta^k \mathbf{s}^{k-1}, \quad k = 1, 2, \dots$$

where

$$\beta^k = \frac{\nabla f^T(\mathbf{x}^k) \nabla f(\mathbf{x}^k)}{\nabla f^T(\mathbf{x}^{k-1}) \nabla f(\mathbf{x}^{k-1})}$$

The one-step BFGS formula is usually more efficient than the Fletcher–Reeves method. It uses somewhat more complex formulas:

$$\begin{aligned} \mathbf{s}^k &= -\nabla f(\mathbf{x}^0) \\ \mathbf{s}^k &= -\nabla f(\mathbf{x}^k) - \frac{(\mathbf{y}^{k-1})^T \nabla f(\mathbf{x}^k)}{(\mathbf{y}^{k-1})^T \mathbf{d}^{k-1}} (\mathbf{y}^{k-1} - \mathbf{d}^{k-1}) \\ &\quad + \frac{(\mathbf{y}^{k-1} - \mathbf{d}^{k-1})^T \mathbf{d}^{k-1} [(\mathbf{y}^{k-1})^T \nabla f(\mathbf{x}^k)]}{[(\mathbf{y}^{k-1})^T \mathbf{d}^{k-1}]^2} \mathbf{y}^{k-1} \quad k = 1, 2, \dots \end{aligned}$$

This formula follows from the BFGS formula for $(\tilde{\mathbf{H}}^k)^{-1}$ by (1) assuming $(\tilde{\mathbf{H}}^{k-1})^{-1} = \mathbf{I}$, (2) computing $(\tilde{\mathbf{H}}^k)^{-1}$ from the update formula, and (3) computing \mathbf{s}^k as $-(\tilde{\mathbf{H}}^k)^{-1} \nabla f(\mathbf{x}^k)$. Both methods minimize a positive-definite quadratic function of n variables in at most n iterations using exact line searches but generally require significantly more iterations than the BFGS procedure for general nonlinear functions. A class of algorithms called variable memory quasi-Newton methods (Nash and Sofer, 1996) partially overcomes this difficulty and provides an effective compromise between standard quasi-Newton and conjugate gradient algorithms.

REFERENCES

- Armijo, L. "Minimization of Functions Having Lipschitz Continuous First Partial Derivatives." *Pac J Math* **16**: 1–3 (1966).
- Avriel, M. *Nonlinear Programming*. Prentice-Hall, Englewood Cliffs, New Jersey (1976).
- Broyden, C. G. "The Convergence of a Class of Double-Rank Minimization Algorithms." *J Inst Math Appl* **6**: 76–90 (1970).
- Dembo, R. S.; S. C. Eisenstat; and T. Steihaug. "Inexact Newton Methods." *SIAM J Num Anal* **19**: 400–408 (1982).
- Dennis, J. E.; and R. B. Schnabel. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. Prentice-Hall, Englewood Cliffs, New Jersey (1996).
- Dixon, L. C. W.; and L. James. "On Stochastic Variable Metric Methods." In *Analysis and Optimization of Stochastic Systems*. Q. L. R. Jacobs et al. eds. Academic Press, London (1980).
- Fletcher, R. "A New Approach to Variable Metric Algorithms." *Comput J* **13**: 317 (1970).
- Fletcher, R. *Practical Methods of Optimization*, vol. 1. John Wiley, New York (1980).
- Fletcher, R.; and C. M. Reeves. "Function Minimization by Conjugate Gradients." *Comput J* **7**: 149–154 (1964).
- Gill, P. E.; W. Murray; and M. H. Wright. *Practical Optimization*. Academic Press, New York (1981).
- Goldfarb, D. "A Family of Variable Metric Methods Derived by Variational Means." *Math Comput* **24**: 23–26 (1970).
- Levenberg, K. "A Method for the Solution of Certain Problems in Least Squares." *Q Appl Math* **2**: 164–168 (1944).

- Marquardt, D. "An Algorithm for Least-Squares Estimation of Nonlinear Parameters." *SIAM J Appl Math* **11**: 431–441 (1963).
- Nash, S. G.; and A. Sofer. *Linear and Nonlinear Programming*. McGraw-Hill, New York (1996).
- Nelder, J. A.; and R. Mead. "A Simplex Method for Function Minimization." *Comput J* **7**: 308–313 (1965).
- Powell, M. J. D. "A New Algorithm for Unconstrained Optimization." In *Nonlinear Programming*. J. B. Rosen; O. L. Mangasarian; and K. Ritter, eds. pp. 31–65, Academic Press, New York (1970).
- Shanno, D. F. "Conditioning of Quasi-Newton Methods for Function Minimization." *Math Comput* **24**: 647–657 (1970).
- Spendley, W.; G. R. Hext; and F. R. Himsworth. "Sequential Application of Simplex Designs in Optimization and Evolutionary Operations." *Technometrics* **4**: 441–461 (1962).
- Uchiyama, T. "Best Size for Refinery and Tankers." *Hydrocarbon Process.* **47**(12): 85–88 (1968).

SUPPLEMENTARY REFERENCES

- Brent, R. P. *Algorithms for Minimization Without Derivatives*. Prentice-Hall, Englewood Cliffs, New Jersey (1973).
- Broyden, C. G. "Quasi-Newton Methods and Their Application to Function Minimization." *Math Comput* **21**: 368 (1967).
- Boggs, P. T.; R. H. Byrd; and R. B. Schnabel. *Numerical Optimization*. SIAM, Philadelphia (1985).
- Hestenes, M. R. *Conjugate-Direction Methods in Optimization*. Springer-Verlag, New York (1980).
- Kelley, C. T. *Iterative Methods for Optimization*. SIAM, Philadelphia (1999).
- Li, J.; and R. R. Rhinehart. "Heuristic Random Optimization." *Comput Chem Engin* **22**: 427–444 (1998).
- Powell, M. J. D. "An Efficient Method for Finding the Minimum of a Function of Several Variables Without Calculating Derivatives." *Comput J* **7**: 155–162 (1964).
- Powell, M. J. D. "Convergence Properties of Algorithms to Nonlinear Optimization." *SIAM Rev* **28**: 487–496 (1986).
- Reklaitis, G. V.; A. Ravindran; and K. M. Ragsdell. *Engineering Optimization—Methods and Applications*. John Wiley, New York (1983).
- Schittkowski, K. *Computational Mathematical Programming*. Springer-Verlag, Berlin (1985).

PROBLEMS

- 6.1** If you carry out an exhaustive search (i.e., examine each grid point) for the optimum of a function of five variables, and each step is $1/20$ of the interval for each variable, how many objective function calculations must be made?

6.2 Consider the following minimization problem:

$$\text{Minimize: } f(\mathbf{x}) = x_1^2 + x_1x_2 + x_2^2 + 3x_1$$

- Find the minimum (or minima) analytically.
 - Are they global or relative minima?
 - Construct four contours of $f(\mathbf{x})$ [lines of constant value of $f(\mathbf{x})$].
 - Is univariate search a good numerical method for finding the optimum of $f(\mathbf{x})$? Why or why not?
 - Suppose the search direction is given by $\mathbf{s} = [1 \ 0]^T$. Start at $(0,0)$, find the optimum point P_1 in that search direction analytically, not numerically. Repeat the exercise for a starting point of $(0,4)$ to find P_2 .
 - Show graphically that a line connecting P_1 and P_2 passes through the optimum.
- 6.3** Determine a regular simplex figure in a three-dimensional space such that the distance between vertices is 0.2 unit and one vertex is at the point $(-1, 2, -2)$.
- 6.4** Carry out the four stages of the simplex method to minimize the function

$$f(\mathbf{x}) = x_1^2 + 3x_2^2$$

starting at $\mathbf{x} = [1 \ 1.5]^T$. Use $\mathbf{x} = [1 \ 2]^T$ for another corner. Show each stage on a graph.

6.5 A three-dimensional simplex optimal search for a minimum provides the following intermediate results:

x vector	Value of objective function
$[0 \ 0 \ 0]^T$	4
$[-\frac{4}{3} \ -\frac{1}{3} \ -\frac{1}{3}]^T$	7
$[-\frac{1}{3} \ -\frac{4}{3} \ -\frac{1}{3}]^T$	10
$[-\frac{1}{3} \ -\frac{1}{3} \ -\frac{4}{3}]^T$	5

What is the next point to be evaluated in the search? What point is dropped?

6.6 Find a direction orthogonal to the vector

$$\mathbf{s} = \left[\frac{1}{\sqrt{3}} \ -\frac{1}{\sqrt{3}} \ -\frac{1}{\sqrt{3}} \right]^T$$

at the point

$$\mathbf{x} = [0 \ 0 \ 0]^T$$

Find a direction conjugate to \mathbf{s} with respect to the Hessian matrix of the objective function $f(\mathbf{x}) = x_1 + 2x_2^2 - x_1x_2$ at the same point.

- 6.7 Given the function $f(\mathbf{x}) = x_1^2 + x_2^2 + 2x_3^2 - x_1x_2$, generate a set of conjugate directions. Carry out two stages of the minimization in the conjugate directions minimizing $f(\mathbf{x})$ in each direction. Did you reach the minimum of $f(\mathbf{x})$? Start at $(1, 1, 1)$.
- 6.8 For what values of \mathbf{x} are the following directions conjugate for the function $f(\mathbf{x}) = x_1^2 + x_1x_2 + 16x_2^2 + x_3^2 - x_1x_2x_3$?

$$\mathbf{s}^{(1)} = \begin{bmatrix} -\frac{1}{\sqrt{3}} \\ \frac{1}{\sqrt{3}} \\ -\frac{1}{\sqrt{3}} \end{bmatrix} \quad \mathbf{s}^{(2)} = \begin{bmatrix} -\frac{1}{\sqrt{3}} \\ \frac{2}{\sqrt{3}} \\ 0 \end{bmatrix}$$

- 6.9 In the minimization of

$$f(\mathbf{x}) = 5x_1^2 + x_2^2 + 2x_1x_2 - 12x_1 - 4x_2 + 8$$

starting at $(0, -2)$, find a search direction \mathbf{s} conjugate to the x_1 axis. Find a second search vector \mathbf{s}_2 conjugate to \mathbf{s}_1 .

- 6.10 (a) Find two directions respectively orthogonal to

$$\mathbf{x}^T = \left[\frac{2}{3}, -\frac{1}{3}, -\frac{2}{3} \right]$$

and each other.

- (b) Find two directions respectively conjugate to the vector in part (a) and to each other for the given matrix

$$\begin{bmatrix} 2 & 1 & 0 \\ 1 & 2 & 1 \\ 0 & 1 & 3 \end{bmatrix}$$

- 6.11 The starting search direction from $\mathbf{x} = [2 \ 2]^T$ to minimize

$$f(\mathbf{x}) = x_1^2 + x_1x_2 + x_2^2 - 3x_1 - 3x_2$$

is the negative gradient. Find a conjugate direction to the starting direction. Is it unique?

- 6.12 Evaluate the gradient of the function

$$f(\mathbf{x}) = (x_1 + x_2)^3 x_3 + x_3^2 x_1^2 x_2^2$$

at the point $\mathbf{x} = [1 \ 1 \ 1]^T$.

6.13 You are asked to maximize

$$f(\mathbf{x}) = x_1 + x_2 - \frac{1}{2}(x_1^2 + 2x_1x_2 + 2x_2^2)$$

Begin at $\mathbf{x} = [1 \ 1]^T$, and select the gradient as the first search direction. Find a second search direction that is conjugate to the first search direction. (Do not continue after getting the second direction.)

6.14 You wish to minimize

$$f(\mathbf{x}) = 10x_1^2 + x_2^2$$

If you use steepest descent starting at $(1, 1)$, will you reach the optimum in

- (a) One iteration
- (b) Two iterations
- (c) More than two?

Explain.

6.15 Evaluate the gradient of the function

$$f(\mathbf{x}) = e^{x_1x_2} - 2e^{x_1} + 2e^{x_2} + (x_1x_2)^2$$

at the point $(0, 0)$.

6.16 Consider minimizing the function $f(\mathbf{x}) = x_1^2 + x_2^2$. Use the formula $\mathbf{x}^{k+1} = \mathbf{x}^k - \alpha \nabla f(\mathbf{x}^k)$, where α is chosen to minimize $f(\mathbf{x})$. Show that \mathbf{x}^{k+1} will be the optimum \mathbf{x} after only one iteration. You should be able to optimize $f(\mathbf{x})$ with respect to α analytically. Start from

$$\mathbf{x}^0 = \begin{bmatrix} 3 \\ 5 \end{bmatrix}$$

6.17 Why is the steepest descent method not widely used in unconstrained optimization codes?

6.18 Use the Fletcher–Reeves search to find the minimum of the objective function

$$(a) f(\mathbf{x}) = 3x_1^2 + x_2^2$$

$$(b) f(\mathbf{x}) = 4(x_1 - 5)^2 + (x_2 - 6)^2$$

starting at $\mathbf{x}^0 = [1 \ 1]^T$.

6.19 Discuss the advantages and disadvantages of the following two search methods for the function shown in Figure P6.19.

- (a) Steepest descent
- (b) Conjugate gradient

Discuss the basic idea behind each of the two methods (don't write out the individual steps, though). Be sure to consider the significance of the starting point for the search.

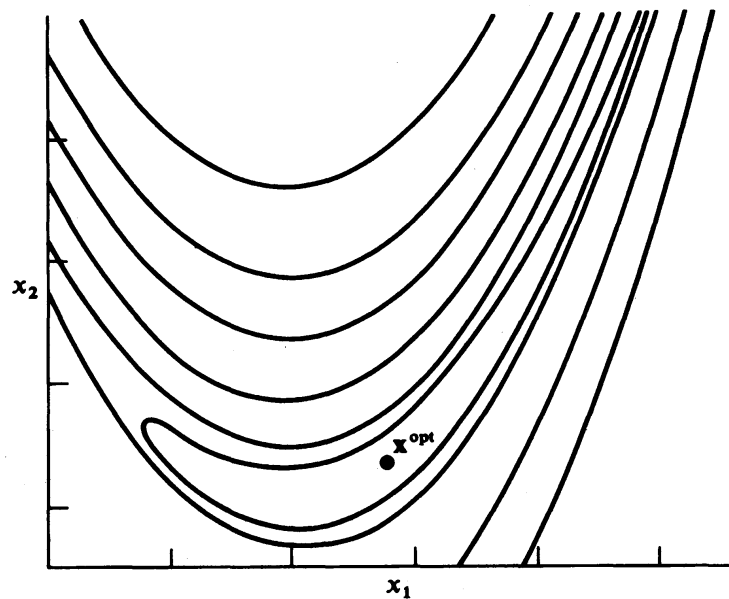


FIGURE P6.19

6.20 Repeat Problem 6.18 for the Woods function.

$$f = \sum_{i=1}^4 F_i(\mathbf{x})$$

$$\text{where } F_1(\mathbf{x}) = -200x_1(x_4 - x_3^2) - (1 - x_1)$$

$$F_2(\mathbf{x}) = 200(x_2 - x_1^2) + 20(x_2 - 1) + 19.8(x_4 - 1)$$

$$F_3(\mathbf{x}) = -180x_3(x_4 - x_3^2) - (1 - x_3)$$

$$F_4(\mathbf{x}) = (-3, -1, -3, -1)$$

6.21 An open cylindrical vessel is to be used to store 10 ft³ of liquid. The objective function for the sum of the operating and capital costs of the vessel is

$$f(h, r) = \frac{1}{\pi r^2 h} 2\pi r h + 10\pi r^2$$

Can Newton's method be used to minimize this function? The solution is $[r^* \ h^*]^T = [0.22 \ 2.16]^T$.

6.22 Is it necessary that the Hessian matrix of the objective function always be positive-definite in an unconstrained minimization problem?

6.23 Cite two circumstances in which the use of the simplex method of multivariate unconstrained optimization might be a better choice than a quasi-Newton method.

6.24 Given the function $f(\mathbf{x}) = 3x_1^2 + 3x_2^2 + 3x_3^2$ to minimize, would you expect that steepest descent or Newton's method (in which adjustment of the step length is used for minimization in the search direction) would be faster in solving the problem from the same starting point $\mathbf{x} = [10 \ 10 \ 10]^T$? Explain the reasons for your answer.

6.25 Consider the following objective functions:

$$(a) \quad f(\mathbf{x}) = 1 + x_1 + x_2 + \frac{4}{x_1} + \frac{9}{x_2}$$

$$(b) \quad f(\mathbf{x}) = (x_1 + 5)^2 + (x_2 + 8)^2 + (x_3 + 7)^2 + 2x_1^2x_2^2 + 4x_1^2x_3^2$$

Will Newton's method converge for these functions?

6.26 Consider the minimization of the objective function

$$f(\mathbf{x}) = x_1^3 + x_1x_2 - x_2^2x_1^2$$

by Newton's method starting from the point $\mathbf{x}^0 = [1 \ 1]^T$. A computer code carefully programmed to execute Newton's method has not been successful. Explain the probable reason(s) for the failure.

6.27 What is the initial direction of search determined by Newton's method for $f(\mathbf{x}) = x_1^2 + 2x_2^2$? What is the step length? How many steps are needed to minimize $f(\mathbf{x})$ analytically?

6.28 Will Newton's method minimize Rosenbrock's function

$$f(\mathbf{x}) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$$

starting at $\mathbf{x}^0 = [-1.2 \ 1.0]^T$ in one stage? How many stages will it take if you minimize $f(\mathbf{x})$ exactly on each stage? How many stages if you let the step length be unity on each stage?

6.29 Find the minimum of the following objective function by (a) Newton's method or (b) Fletcher-Reeves conjugate gradient

$$f(\mathbf{x}) = 8x_1^2 + 4x_1x_2 + 5x_2^2.$$

starting at $\mathbf{x}^T = [10 \ 10]$.

6.30 Solve the following problems by Newton's method:

Minimize:

$$(a) \quad f(\mathbf{x}) = 1 + x_1 + x_2 + x_3 + x_4 + x_1x_2 + x_1x_3 + x_1x_4 \\ + x_2x_3 + x_2x_4 + x_3x_4 + x_1^2 + x_2^2 + x_3^2 + x_4^2$$

starting from

$$\mathbf{x}^0 = [-3 \ -30 \ -4 \ -0.1]^T \quad \text{and also} \quad \mathbf{x}^0 = [0.5 \ 1.0 \ 8.0 \ -0.7]^T$$

$$(b) \quad f(\mathbf{x}) = x_1x_2^2x_3^3x_4^4 [\exp - (x_1 + x_2 + x_3 + x_4)]$$

starting from

$$\mathbf{x}^0 = [3 \ 4 \ 0.5 \ 1]^T$$

- 6.31** List the relative advantages and disadvantages (there can be more than one) of the following methods for a two-variable optimization problem such as Rosenbrock's "banana" function (see Fig. P6.19)
- (a) Sequential simplex
 - (b) Conjugate gradient
 - (c) Newton's method

Would your evaluation change if there were 20 independent variables in the optimization problem?

- 6.32** Find the maximum of the function $f(\mathbf{x}) = 100 - (10 - x_1)^2 - (5 - x_2)^2$ by the
- (a) Simplex method
 - (b) Newton's method
 - (c) BFGS method

Start at $\mathbf{x}^T = [0 \ 0]$. Show all equations and intermediate calculations you use. For the simplex method, carry out only five stages of the minimization.

- 6.33** For the function $f(\mathbf{x}) = (x - 100)^2$, use
- (a) Newton's method
 - (b) Quasi-Newton method
 - (c) Quadratic interpolation

to minimize the function. Show all equations and intermediate calculations you use. Start at $\mathbf{x} = 0$.

- 6.34** For the function $f(\mathbf{x}) = (x - 100)^3$, use
- (a) Steepest descent
 - (b) Newton's method
 - (c) Quasi-Newton method
 - (d) Quadratic interpolation

to minimize the function. Show all equations and intermediate calculations you use. Start at $\mathbf{x} = 0$.

- 6.35** How can the inverse of the Hessian matrix for the function

$$f(\mathbf{x}) = 2x_1^2 - 2x_2^2 - x_1x_2$$

be approximated by a positive-definite matrix using the method of Marquardt?

- 6.36** You are to minimize $f(\mathbf{x}) = 2x_1^2 - 4x_1x_2 + x_2^2$. Is $\mathbf{H}(\mathbf{x})$ positive-definite? If not, start at $\mathbf{x}^0 = [2 \ 2]^T$, and develop an approximation of $\mathbf{H}(\mathbf{x})$ that is positive-definite by Marquardt's method.

- 6.37** Show how to make the Hessian matrix of the following objective function positive-definite at $\mathbf{x} = [1 \ 1]^T$ by using Marquardt's method:

$$f(\mathbf{x}) = 2x_1^3 - 6x_1x_2 + x_2^2$$

6.38 The Hessian matrix of the following function

$$f(\mathbf{x}) = u_1^2 + u_2^2 + u_3^2$$

where $u_1 = 1.5 - x_1(1 - x_2)$

$$u_2 = 2.25 - x_1(1 - x_2^2)$$

$$u_3 = 2.625 - x_1(1 - x_2^3)$$

is not positive-definite in the vicinity of $\mathbf{x} = [0 \ 1]^T$ and Newton's method will terminate at a saddle point if started there. If you start at $\mathbf{x} = [0 \ 1]^T$, what procedure should you carry out to make a Newton or quasi-Newton method continue with searches to reach the optimum, which is in the vicinity of $\mathbf{x} = [3 \ 0.5]^T$?

6.39 Determine whether the following statements are true or false, and explain the reasons for your answer.

- All search methods based on conjugate directions (e.g., Fletcher-Reeves method) always use conjugate directions.
- The matrix, or its inverse, used in the BFGS relation, is an approximation of the Hessian matrix, or its inverse, of the objective function $[\nabla^2 f(\mathbf{x})]$.
- The BFGS version has the advantage over a pure Newton's method in that the latter requires second derivatives, whereas the former requires only first derivatives to get the search direction.

6.40 For the quasi-Newton method discussed in Section 6.4, give the values of the elements of the approximate to the Hessian (inverse Hessian) matrix for the first two stages of search for the following problems:

(a) Maximize: $f(\mathbf{x}) = -x_1^2 + x_1 - x_2^2 + x_2 + 4$

(b) Minimize: $f(\mathbf{x}) = x_1^3 \exp[x_2 - x_1^2 - 10(x_1 - x_2)^2]$

$$f(\mathbf{x}) = x_1^2 + x_2^2 + x_3^2 + x_4^2$$

starting from the point (1, 1) or (1, 1, 1, 1) as the case may be.

6.41 Estimate the values of the parameters k_1 and k_2 by minimizing the sum of the squares of the deviations

$$\phi = \sum_{i=1}^n (y_{\text{observed}} - y_{\text{predicted}})_i^2$$

where

$$y_{\text{predicted}} = \frac{k_1}{k_1 - k_2} (e^{-k_2 t} - e^{-k_1 t})$$

for the following data:

t	y_{observed}
0.5	0.263
1.0	0.455
1.5	0.548

Plot the sum-of-squares surface with the estimated coefficients.

6.42 Repeat Problem 6.41 for the following model and data:

$$y = \frac{k_1 x_1}{1 + k_2 x_1 + k_3 x_2}$$

y_{observed}	x_1	x_2
0.126	1	1
0.219	2	1
0.076	1	2
0.126	2	2
0.186	0.1	0

6.43 Approximate the minimum value of the integral

$$\int_0^1 \left[\left(\frac{dy}{dx} \right)^2 - 2yx^2 \right] dx$$

subject to the boundary conditions $dy/dx = 0$ at $x = 0$ and $y = 0$ at $x = 1$.

Hint: Assume a trial function $y(x) = a(1 - x^2)$ that satisfies the boundary conditions and find the value of a that minimizes the integral. Will a more complicated trial function that satisfies the boundary conditions improve the estimate of the minimum of the integral?

6.44 In a decision problem it is desired to minimize the expected risk defined as follows:

$$\varepsilon\{\text{risk}\} = (1 - P)c_1[1 - F(b)] + Pc_2\theta \left(\frac{b}{2} + \frac{2\pi}{4} \right) F \left(\frac{b}{2} - \frac{\sqrt{2\pi}}{4} \right)$$

where $F(b) = \int_{-\infty}^b e^{-u^2/2\theta^2} du$ (normal probability function)

$$c_1 = 1.25 \times 10^5$$

$$c_2 = 15$$

$$\theta = 2000$$

$$P = 0.25$$

Find the minimum expected risk and b .

6.45 The function

$$f(\mathbf{x}) = (1 + 8x_1 - 7x_1^2 + \frac{7}{3}x_1^3 - \frac{1}{4}x_1^4)(x_2^2 e^{-x_2})F(\mathbf{x}_3)$$

has two maxima and a saddle point. For (a) $F(\mathbf{x}_3) = 1$ and (b) $F(\mathbf{x}_3) = x_3 e^{-(x_3+1)}$, locate the global optimum by a search technique.

Answer: (a) $\mathbf{x}^* = [4 \ 2]^T$ and (b) $\mathbf{x}^* = [4 \ 2 \ 1]^T$.

- 6.46** By starting with (a) $\mathbf{x}^0 = [2 \ 1]^T$ and (b) $\mathbf{x}^0 = [2 \ 1 \ 1]^T$, can you reach the solution for Problem 6.45? Repeat for (a) $\mathbf{x}^0 = [2 \ 2]^T$ and (b) $\mathbf{x}^0 = [2 \ 2 \ 1]^T$.

Hint: $[2 \ 2 \ 1]$ is a saddle point.

- 6.47** Estimate the coefficients in the correlation

$$y = ax_1^{b_1}x_2^{b_2}$$

from the following experimental data by minimizing the sum of the square of the deviations between the experimental and predicted values of y .

y_{expt}	x_1	x_2
46.5	2.0	36.0
591	6.0	8.0
1285	9.0	3.0
36.8	2.5	6.25
241	4.5	7.84
1075	9.5	1.44
1024	8.0	4.0
151	4.0	7.0
80	3.0	9.0
485	7.0	2.0
632	6.5	5.0

- 6.48** The cost of refined oil when shipped via the Malacca Straits to Japan in dollars per kiloliter was given (Uchiyama, 1968) as the linear sum of the crude oil cost, the insurance, customs, freight cost for the oil, loading and unloading cost, sea berth cost, submarine pipe cost, storage cost, tank area cost, refining cost, and freight cost of products as

$$\begin{aligned}
 c = c_c + c_i + c_x + & \frac{2.09 \times 10^4 t^{-0.3017}}{360} + \frac{1.064 \times 10^6 at^{0.4925}}{52.47q(360)} \\
 & + \frac{4.242 \times 10^4 at^{0.7952} + 1.813ip(nt + 1.2q)^{0.861}}{52.47q(360)} \\
 & + \frac{4.25 \times 10^3 a(nt + 1.2q)}{52.47q(360)} + \frac{5.042 \times 10^3 q^{-0.1899}}{360} \\
 & + \frac{0.1049q^{0.671}}{360}
 \end{aligned}$$

where a = annual fixed charges, fraction (0.20)

c_c = crude oil price, \$/kL (12.50)

c_i = insurance cost, \$/kL (0.50)

c_x = customs cost, \$/kL (0.90)

i = interest rate (0.10)

- n = number of ports (2)
- p = land price, $\$/\text{m}^2$ (7000)
- q = refinery capacity, bbl/day
- t = tanker size, kL

Given the values indicated in parentheses, use a computer code to compute the minimum cost of oil and the optimum tanker size t and refinery size q by Newton's method and the quasi-Newton method (note that 1 kL = 6.29 bbl).

(The answers in the reference were

$$t = 427,000 \text{ dwt} \approx 485,000 \text{ kL}$$

$$q = 185,000 \text{ bbl/day})$$

LINEAR PROGRAMMING (LP) AND APPLICATIONS

7.1	Geometry of Linear Programs	223
7.2	Basic Linear Programming Definitions and Results	227
7.3	Simplex Algorithm	233
7.4	Barrier Methods	242
7.5	Sensitivity Analysis	242
7.6	Linear Mixed Integer Programs	243
7.7	LP Software	243
7.8	A Transportation Problem Using the EXCEL Solver	
	Spreadsheet Formulation	245
7.9	Network Flow and Assignment Problems	252
	References	253
	Supplementary References	253
	Problems	254

LINEAR PROGRAMMING (LP) IS one of the most widely used optimization techniques and perhaps the most effective. The term *linear programming* was coined by George Dantzig in 1947 to refer to problems in which both the objective function and the constraints are linear (Dantzig, 1998; Martin, 1999; Vanderbei, 1999). The word *programming* does not refer to computer programming, but means optimization. This is also true in the phrases “nonlinear programming,” “integer programming,” and so on. The following are examples of LP that occur in plant management:

1. Assign employees to schedules so that the workforce is adequate each day of the week and worker satisfaction and productivity are as high as possible.
2. Select products to manufacture in the upcoming period, taking best advantage of existing resources and current prices to yield maximum profit.
3. Find a pattern of distribution from plants to warehouses that will minimize costs within the capacity limitations.
4. Submit bids on procurement contracts to take into account profit, competitors' bids, and operating constraints.

When stated mathematically, each of these problems potentially involves many variables, many equations, and many inequalities. A solution must not only satisfy all of the constraints, but also must achieve an extremum of the objective function, such as maximizing profit or minimizing cost. With the aid of modern software you can formulate and solve LP problems with many thousands of variables and constraints.

7.1 GEOMETRY OF LINEAR PROGRAMS

Consider the problem

$$\begin{aligned} \text{Maximize: } f &= x_1 + 3x_2 \\ \text{Subject to: } & -x_1 + x_2 \leq 1 \\ & x_1 + x_2 \leq 2 \\ & x_1 \geq 0, \quad x_2 \geq 0 \end{aligned} \tag{7.1}$$

The feasible region lies within the unshaded area of Figure 7.1 defined by the intersections of the half spaces satisfying the linear inequalities. The numbered points are called extreme points, corner points, or vertices of this set. If the constraints are linear, only a finite number of vertices exist.

Contours of constant value of the objective function f are defined by the linear equation

$$x_1 + 3x_2 = \text{Constant} = c \tag{7.2}$$

As c varies, the contour is moved parallel to itself. The maximum value of f is the largest c for which the line has at least one point in common with the constraint set.

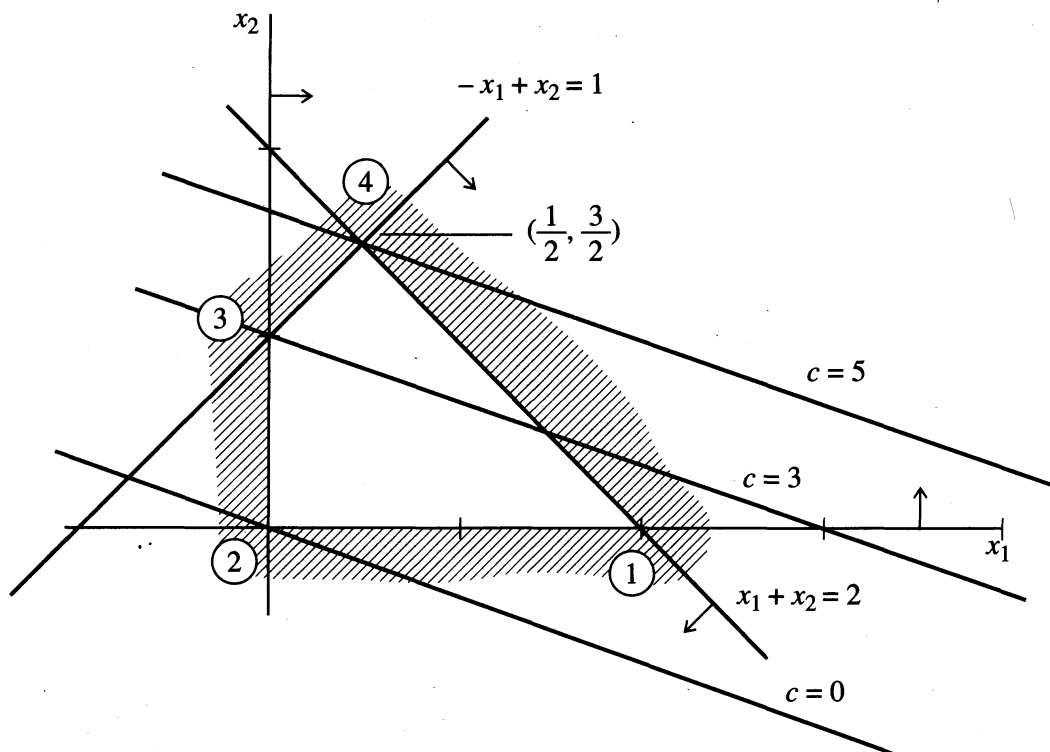


FIGURE 7.1
Geometry of a linear program.

For Figure 7.1, this point occurs for $c = 5$, and the optimal values of \mathbf{x} are $x_1 = 0.5$, $x_2 = 1.5$. Note that the maximum value occurs at a vertex of the constraint set. If the problem seeks to minimize f , the minimum is at the origin, which is again a vertex. If the objective function were $f = 2x_1 + 2x_2$, the line $f = \text{Constant}$ would be parallel to one of the constraint boundaries, $x_1 + x_2 = 2$. In this case the maximum occurs at two extreme points, $(x_1 = 0.5, x_2 = 1.5)$ and $(x_1 = 2, x_2 = 0)$ and, in fact, also occurs at all points on the line segment joining these vertices.

Two additional cases can exist. First, if the constraint $x_1 + x_2 \leq 2$ had been removed, the feasible region would appear as in Figure 7.2, that is, the set would be unbounded. Then $\max f$ is also unbounded because f can be made as large as desired subject to the constraints. Second, at the opposite extreme, the constraint set could be empty, as in the case where $x_1 + x_2 \leq 2$ is replaced by $x_1 + x_2 \leq -1$. Thus an LP problem may have (1) no solution, (2) an unbounded solution, (3) a single optimal solution, or (4) an infinite number of optimal solutions. The methods to be developed deal with all these possibilities.

The fact that the extremum of a linear program always occurs at a vertex of the feasible region is the single most important property of linear programs. It is true for any number of variables (i.e., more than two dimensions) and forms the basis for the simplex method for solving linear programs (not to be confused with the simplex method discussed in Section 6.1.4).

Of course, for many variables the geometrical ideas used here cannot be visualized, and therefore the extreme points must be characterized algebraically. This is

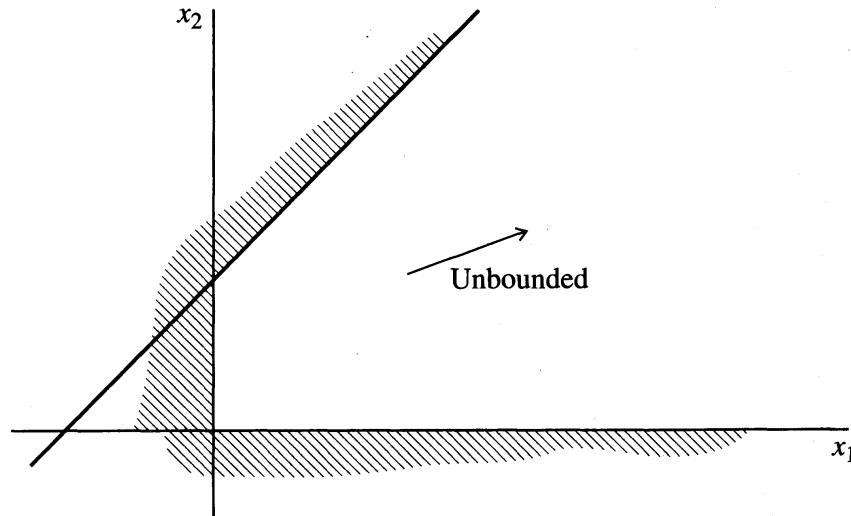


FIGURE 7.2
Unbounded minimum.

done in the next two sections, in which the problem is placed in standard form and the basic results of linear programming are stated.

Standard form for linear programs

An LP problem can always be written in the following form. Choose $\mathbf{x} = (x_1, x_2, \dots, x_n)$ to minimize

$$f = \sum_{j=1}^n c_j x_j \quad (7.3)$$

$$\text{Subject to: } \sum_{j=1}^n a_{ij} x_j = b_i, \quad i = 1, 2, \dots, m \quad (7.4)$$

$$l_j \leq x_j \leq u_j, \quad j = 1, \dots, n \quad (7.5)$$

where c_j are the n objective function coefficients, a_{ij} and b_i are parameters in the m linear equality constraints, and l_j and u_j are lower and upper bounds with $l_j \leq u_j$. Both l_j and u_j may be positive or negative. In matrix form, this problem is

$$\text{Minimize: } f = \mathbf{c}\mathbf{x} \quad (7.6)$$

$$\text{Subject to: } \mathbf{A}\mathbf{x} = \mathbf{b} \quad (7.7)$$

$$\text{and } \mathbf{l} \leq \mathbf{x} \leq \mathbf{u} \quad (7.8)$$

\mathbf{A} is an $m \times n$ matrix whose (i, j) element is the constraint coefficient a_{ij} , and \mathbf{c} , \mathbf{b} , \mathbf{l} , \mathbf{u} are vectors whose components are c_j , b_j , l_j , u_j , respectively. If any of the Equations (7.7) were redundant, that is, linear combinations of the others, they could be deleted without changing any solutions of the system. If there is no solution, or if there is only one solution for Equation (7.7), there can be no optimization. Thus the

case of greatest interest is where the system of equations (7.7) has more unknowns than equations and has at least two and potentially an infinite number of solutions. This occurs if and only if

$$n > m$$

and

$$\text{Rank}(\mathbf{A}) = m$$

We assume these conditions are true in what follows. The problem of linear programming is to first detect whether solutions exist, and, if so, to find one yielding the minimum f .

Note that all the constraints in Equation (7.4) are equalities. It is necessary to place the problem in this form to solve it most easily (equations are easier to work with here than inequalities). If the original system is not of this form, it may easily be transformed by use of so-called *slack variables*. If a given constraint is an inequality, for example,

$$\sum_{j=1}^n a_{ij}x_j \leq b_i$$

then define a slack variable $x_{n+i} \geq 0$ such that

$$\sum_{j=1}^n a_{ij}x_j + x_{n+i} = b_i$$

and the inequality becomes an equality. Similarly, if the inequality is

$$\sum_{j=1}^n a_{ij}x_j \geq b_i$$

we write

$$\sum_{j=1}^n a_{ij}x_j - x_{n+i} = b_i$$

Note that the slacks must be nonnegative to guarantee that the inequalities are satisfied.

EXAMPLE 7.1 STANDARD LP FORM

Transform the following linear program into standard form:

$$\begin{aligned} \text{Minimize: } & f = x_1 + x_2 \\ \text{Subject to: } & 2x_1 + 3x_2 \leq 6 \\ & x_1 + 7x_2 \geq 4 \\ & x_1 + x_2 = 3 \\ & x_1 \geq 0, \quad x_2 \text{ unconstrained in sign} \end{aligned}$$

Solution. Define slack variables $x_3 \geq 0, x_4 \geq 0$. Then the problem becomes

$$\begin{aligned} \text{Minimize: } f &= x_1 + x_2 \\ \text{Subject to: } 2x_1 + 3x_2 + x_3 &= 6 \\ x_1 + 7x_2 - x_4 &= 4 \\ x_1 + x_2 &= 3 \\ x_1 \geq 0, \quad x_3 \geq 0, \quad x_4 &\geq 0 \end{aligned}$$

In the rest of this chapter, we assume that the rows of the constraint matrix \mathbf{A} are linearly independent, that is, $\text{rank}(\mathbf{A}) = m$. If a slack variable is inserted in every row, then \mathbf{A} contains a submatrix that is the identity matrix. In the preceding example, if we insert a slack variable x_5 into the equality:

$$\begin{aligned} x_1 + x_2 + x_5 &= 3 \\ 0 \leq x_5 \leq 0 \quad (\text{i.e., } x_5 &= 0) \end{aligned}$$

then the rows of \mathbf{A} are independent. Modern LP solvers automatically transform problems in this way.

7.2 BASIC LINEAR PROGRAMMING DEFINITIONS AND RESULTS

We now generalize the ideas illustrated earlier from 2 to n dimensions. Proofs of the following theorems may be found in Dantzig (1963). First a number of standard definitions are given.

DEFINITION 1. A *feasible solution* to the linear programming problem is a vector $\mathbf{x} = (x_1, x_2, \dots, x_n)$ that satisfies Equations (7.7) and the bounds (7.8).

DEFINITION 2. A *basis matrix* is an $m \times m$ nonsingular matrix formed from some m columns of the constraint matrix \mathbf{A} (*Note:* Because $\text{rank}(\mathbf{A}) = m$, \mathbf{A} contains at least one basis matrix).

DEFINITION 3. A *basic solution* to a linear program is the unique vector determined by choosing a basis matrix, setting each of the $n - m$ variables associated with columns of \mathbf{A} not in the basis matrix equal to either l_j or u_j , and solving the resulting square, nonsingular system of equations for the remaining m variables.

DEFINITION 4. A *basic feasible solution* is a basic solution in which all variables satisfy their bounds (7.8).

DEFINITION 5. A *nondegenerate basic feasible solution* is a basic feasible solution in which all basic variables x_j are strictly between their bounds, that is, $l_j < x_j < u_j$.

DEFINITION 6. An *optimal solution* is a feasible solution that also minimizes f in Equation (7.6).

For example, in the system

$$\begin{aligned} -x_1 + x_2 + x_3 &= 1 \\ x_1 + x_2 + x_4 &= 2 \\ x_i &\geq 0, \quad i = 1, \dots, 4 \end{aligned} \tag{7.9}$$

obtained from Equation (7.1) by adding slack variables x_3 and x_4 , the matrix

$$\mathbf{B} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

formed from columns 3 and 4 of the equations in (7.9) is nonsingular and hence is a basis matrix. The corresponding basic solution of (7.9)

$$x_1 = 0, \quad x_2 = 0, \quad x_3 = 1, \quad x_4 = 1$$

is a nondegenerate basic feasible solution. The matrix

$$\mathbf{B}_1 = \begin{bmatrix} -1 & 0 \\ 1 & 1 \end{bmatrix}$$

formed from columns 1 and 4 of Equation (7.9) is also a basis matrix. The corresponding basic solution is obtained by setting $x_2 = x_3 = 0$ and solving

$$\begin{aligned} -x_1 &= 1 \\ x_1 + x_4 &= 2 \end{aligned}$$

yielding $x_1 = -1$, $x_4 = 3$. This basic solution is not feasible.

The importance of these definitions is brought out by the following results:

RESULT 1. The objective function f assumes its minimum at a vertex of the feasible region. If it assumes its minimum at more than one vertex, then it takes on the same value at every point of the line segment joining any two optimal vertices.

This theorem is a multidimensional generalization of the geometric arguments given previously. By result 1, in searching for a solution, we need only look at vertices. It is thus of interest to know how to characterize vertices in many dimensions algebraically. This information is given by the next result.

RESULT 2. A vector $\mathbf{x} = (x_1, \dots, x_n)$ is a vertex of the constraint set of an LP problem if and only if \mathbf{x} is a basic feasible solution of the constraints (7.7)–(7.8).

Result 2 is true in two dimensions as can be seen from the example of relations (7.1), whose constraints have been rewritten in equation form in (7.9). The (x_1, x_2) coordinates of the vertex at $x_1 = 0$, $x_2 = 1$ are given by the (x_1, x_2) coordinates of the basic feasible solution

$$x_1 = 0, \quad x_2 = 1, \quad x_3 = 0, \quad x_4 = 1$$

The optimal vertex corresponds to the basic feasible solution

$$x_1 = 0.5, \quad x_2 = 1.5, \quad x_3 = x_4 = 0$$

An alternative definition of a vertex provides geometric insight and generalizes easily to nonlinear problems. Refer again to Figure 7.1. There are two variables, and each vertex is at the intersection of two *active constraints*. If there were three variables, active constraints would correspond to planes, and vertices would be determined by the intersection of at least three active constraints. For n variables, at least n hyperplanes must interact to define a point. We say “at least,” because it is possible that more than n hyperplanes pass through a vertex. One can always draw other redundant constraints through the vertices in Figure 7.1.

We can state these ideas precisely as follows. Consider any optimization problem with n variables, let x be any feasible point, and let $n_{\text{act}}(\mathbf{x})$ be the number of active constraints at \mathbf{x} . Recall that a constraint is active at \mathbf{x} if it holds as an equality there. Hence equality constraints are active at any feasible point, but an inequality constraint may be active or inactive. Remember to include simple upper or lower bounds on the variables when counting active constraints. We define the *number of degrees of freedom* (dof) at \mathbf{x} as

$$\text{dof}(\mathbf{x}) = n - n_{\text{act}}(\mathbf{x})$$

DEFINITION: A feasible point \mathbf{x} is called a *vertex* if $\text{dof}(\mathbf{x}) \leq 0$ and the coefficient matrix of the active constraints at \mathbf{x} has rank n . It is a *nondegenerate* vertex if $\text{dof}(\mathbf{x}) = 0$, and a *degenerate* vertex if $\text{dof}(\mathbf{x}) < 0$, in which case $\text{abs}[\text{dof}(\mathbf{x})]$ is called the *degree of degeneracy* at \mathbf{x} .

Comparing this definition with the previous one (\mathbf{x} is a vertex if and only if it is a basic feasible solution), if \mathbf{x} is a basic feasible solution, then $n - m$ nonbasic bounds are active, plus m equalities, so

$$n_{\text{act}}(\mathbf{x}) \geq n - m + m = n$$

and $\text{dof}(\mathbf{x}) \leq 0$. If k basic variables are at their bounds, $n_{\text{act}}(\mathbf{x}) = n + k$, and \mathbf{x} is a degenerate vertex with degree of degeneracy k . It is straightforward to show that the active constraint matrix has rank n . One can reverse the argument, showing the definitions are equivalent.

In nonlinear programming problems, optimal solutions need not occur at vertices and can occur at points with positive degrees of freedom. It is possible to have no active constraints at a solution, for example in unconstrained problems. We consider nonlinear problems with constraints in Chapter 8.

Results 1 and 2 imply that, in searching for an optimal solution, we need only consider vertices, hence only basic feasible solutions. Because a basic feasible solution has m basic variables, an upper bound to the number of basic feasible solutions is the number of ways m variables can be selected from a group of n variables, which is

$$\binom{n}{m} = \frac{n!}{(n-m)! m!}$$

For large n and m this is a very large number. Thus, for large problems, it is impossible to evaluate f at all vertices to find the minimum. What is needed is a computational

scheme that selects, in an orderly fashion, a sequence of vertices, each one yielding a lower value of f , until finally the minimum is attained. In this way we consider only a small subset of the vertices. The simplex method, devised by G. B. Dantzig, is such a scheme. This procedure finds a vertex and determines whether it is optimal. If not, it finds a neighboring vertex at which the value of f is less than or equal to the previous value. The process is iterated and in a finite number of steps (usually between m and $2m$) the minimum is found. The simplex method also discovers whether the problem has no finite minimal solution (i.e., $\min f = -\infty$) or if it has no feasible solutions (i.e., an empty constraint set). It is a powerful scheme for solving any linear programming problem.

To explain the method, it is necessary to know how to go from one basic feasible solution (BFS) to another, how to identify an optimal BFS, and how to find a better BFS from a BFS that is not optimal. We consider these questions in the following two sections. The notation and approach used is that of Dantzig (1998).

Systems of linear equations and equivalent systems

Consider the system of m linear equations in n unknowns

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &= b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n &= b_2 \\ \vdots & \\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n &= b_m \end{aligned} \tag{7.10}$$

A solution to this system is any set of variables $x_1 \dots x_n$ that simultaneously satisfies all equations. The set of all solutions to the system is called its solution set. The system may have one, many, or no solutions. If there is no solution, the equations are said to be inconsistent, and their solution set is empty.

Equivalent systems and elementary operations

Two systems of equations are said to be equivalent if they have the same solution sets. Dantzig (1998) proved that the following operations transform a given linear system into an equivalent system:

1. Multiplying any equation E_i by a constant $q \neq 0$
2. Replacing any equation E_i by the equation $E_i + qE_j$, where E_j is any other equation of the system

These operations are called elementary row operations. For example, the linear system of Equations (7.9)

$$\begin{aligned} -x_1 + x_2 + x_3 &= 1 \\ x_1 + x_2 + x_4 &= 2 \end{aligned}$$

may be transformed into an equivalent system by multiplying the first equation by -1 and adding it to the second, yielding

$$\begin{aligned} -x_1 + x_2 + x_3 &= 1 \\ 2x_1 - x_3 + x_4 &= 1 \end{aligned}$$

Note that the solution $x_1 = 0$, $x_3 = 0$, $x_2 = 1$, $x_4 = 2$ is a solution of both systems. In fact, any solution of one system is a solution of the other.

Pivoting

A particular sequence of elementary row operations finds special application in linear programming. This sequence is called a *pivot operation*, defined as follows.

DEFINITION. A pivot operation consists of m elementary operations that replace a linear system by an equivalent system in which a specified variable has a coefficient of unity in one equation and zero elsewhere. The detailed steps are as follows:

1. Select a term $a_{rs}x_s$, in row (equation) r , column (variable) s , with $a_{rs} \neq 0$ called the pivot term.
2. Replace the r th equation E_r by the r th equation multiplied by $1/a_{rs}$.
3. For each $i = 1, 2, \dots, m$ except $i = r$, replace the i th equation E_i by $E_i - a_{is}/a_{rs}E_r$, that is, by the sum of E_i and the replaced r th equation multiplied by $-a_{is}$.

EXAMPLE 7.2 USE OF PIVOT OPERATIONS

Consider the system

$$2x_1 + 3x_2 - 4x_3 + x_4 = 1 \quad (a)$$

$$x_1 - x_2 + 5x_4 = 6 \quad (b)$$

$$3x_1 + x_2 + x_3 = 2 \quad (c)$$

Transform the set of equations to an equivalent system in which x_1 is eliminated from all but Equation (a), but having a unity coefficient in Equation (a).

Solution. Choose the term $2x_1$ as the pivot term. The first operation is to make the coefficient of this term unity, so we divide Equation (a) by 2, yielding the equivalent system

$$x_1 + 1.5x_2 - 2x_3 + 0.5x_4 = 0.5 \quad (a')$$

$$x_1 - x_2 + 5x_4 = 6 \quad (b)$$

$$3x_1 + x_2 + x_3 = 2 \quad (c)$$

The next operation eliminates x_1 from Equation (b) by multiplying (a') by -1 and adding the result to Equation (b), yielding

$$x_1 + 1.5x_2 - 2x_3 + 0.5x_4 = 0.5 \quad (a')$$

$$-2.5x_2 + 2x_3 + 4.5x_4 = 5.5 \quad (b')$$

$$3x_1 + x_2 + x_3 = 2 \quad (c)$$

Finally, we eliminate x_1 from Equation (c) by multiplying (a') by -3 and adding the result to Equation (c), yielding

$$x_1 + 1.5x_2 - 2x_3 + 0.5x_4 = 0.5 \quad (a')$$

$$-2.5x_2 + 2x_3 + 4.5x_4 = 5.5 \quad (b')$$

$$3.5x_2 + 7x_3 - 1.5x_4 = 0.5 \quad (c')$$

Canonical systems

In the following discussion we assume that, in the system of Equations (7.6)–(7.8), all lower bounds $l_j = 0$, and all upper bounds $u_j = +\infty$, that is, that the bounds become $x \geq 0$. This simplifies the exposition. The simplex method is readily extended to general bounds [see Dantzig (1998)]. Assume that the first m columns of the linear system (7.7) form a basis matrix \mathbf{B} . Multiplying each column of (7.7) by \mathbf{B}^{-1} yields a transformed (but equivalent) system in which the coefficients of the variables (x_1, \dots, x_m) are an identity matrix. Such a system is called *canonical* and has the form shown in Table 7.1.

The variables x_1, \dots, x_m are associated with the columns of \mathbf{B} and are called basic variables. They are also called dependent, because if values are assigned to the nonbasic, or independent variables, x_{m+1}, \dots, x_n , then x_1, \dots, x_m can be determined immediately. In particular, if x_{m+1}, \dots, x_n are all assigned zero values then we obtain the basic solution

$$x_1 = \bar{b}_1, x_2 = \bar{b}_2, \dots, x_m = \bar{b}_m, x_{m+1} = x_{m+2} = \dots = x_n = 0$$

TABLE 7.1
Canonical system with basic variables x_1, x_2, \dots, x_m

Dependent (basic) variables	Independent (nonbasic) variables	Constants
x_1	$+\bar{a}_{1,m+1}x_{m+1} + \bar{a}_{1,m+2}x_{m+2} + \dots + \bar{a}_{1n}x_n$	$= \bar{b}_1$
x_2	$+\bar{a}_{2,m+1}x_{m+1} + \bar{a}_{2,m+2}x_{m+2} + \dots + \bar{a}_{2n}x_n$	$= \bar{b}_2$
\vdots	\vdots	\vdots
x_m	$+\bar{a}_{m,m+1}x_{m+1} + \bar{a}_{m,m+2}x_{m+2} + \dots + \bar{a}_{mn}x_n$	$= \bar{b}_m$

If

$$\bar{b}_i \geq 0, \quad i = 1, \dots, m$$

then this is a basic feasible solution. If one or more $\bar{b}_i = 0$, the basic feasible solution is degenerate.

Instead of actually computing \mathbf{B}^{-1} and multiplying the linear system (7.7) by it, we can place Equation (7.7) in canonical form by a sequence of m pivot operations. First pivot on the term $a_{11}x_1$ if $a_{11} \neq 0$. If $a_{11} = 0$, there exists an element in its first row that is nonzero, since \mathbf{B} is nonsingular. Rearranging the columns makes this the (1, 1) element and allows the pivot. Repeating this procedure for the terms $a_{22}x_2, \dots, a_{mm}x_m$ generates the canonical form. Such a form will be used to begin the simplex algorithm.

7.3 SIMPLEX ALGORITHM

The simplex method is a two-phase procedure for finding an optimal solution to LP problems. Phase 1 finds an initial basic feasible solution if one exists or gives the information that one does not exist (in which case the constraints are inconsistent and the problem has no solution). Phase 2 uses this solution as a starting point and either (1) finds a minimizing solution or (2) yields the information that the minimum is unbounded (i.e., $-\infty$). Both phases use the simplex algorithm described here.

In initiating the simplex algorithm, we treat the objective function

$$f = c_1x_1 + c_2x_2 + \dots + c_nx_n$$

as just another equation, that is,

$$-f + c_1x_1 + c_2x_2 + \dots + c_nx_n = 0 \quad (7.11)$$

which we include in the set to form an augmented system of equations. The simplex algorithm is always initiated with this augmented system in canonical form. The basic variables are some m of the x 's, which we renumber to make the first m , that is, $x_1 \dots x_m$ and $-f$. The problem can then be stated as follows.

Find values of $x_1 \geq 0, x_2 \geq 0, \dots, x_n \geq 0$ and $\min f$ satisfying

$$\begin{array}{rcccc} x_1 & & + \bar{a}_{1,m+1}x_{m+1} + \dots + \bar{a}_{1n}x_n & = & \bar{b}_1 \\ & x_2 & & & \\ & \vdots & & & \\ & & \vdots & & \\ & & & & \\ & x_m & + \bar{a}_{m,m+1}x_{m+1} + \dots + \bar{a}_{mn}x_n & = & \bar{b}_m \\ & & & & \\ & (-f) & + \bar{c}_{m+1}x_{m+1} + \dots + \bar{c}_nx_n & = & -\bar{f} \end{array} \quad (7.12)$$

In this canonical form the basic solution is

$$f = \bar{f}, \bar{x}_1 = \bar{b}_1, \dots, \bar{x}_m = \bar{b}_m, x_{m+1} = x_{m+2} = \dots = x_n = 0 \quad (7.13)$$

We assume that this basic solution is feasible, that is,

$$\bar{b}_1 \geq 0, \bar{b}_2 \geq 0, \dots, \bar{b}_m \geq 0 \quad (7.14)$$

The workings of phases 1 and 2 guarantee that this assumption is always satisfied. If Equation (7.14) holds, we say that the linear programming problem is in feasible canonical form.

Test for optimality

If the problem is in feasible canonical form, we have a vertex directly at hand, represented by the basic feasible solution (7.13). But the form provides even more valuable information. By merely glancing at the numbers $\bar{c}_j, j = m + 1, \dots, n$, you can tell if this extreme point is optimal and, if not, you can move to a better one. Consider first the optimality test, given by the following result.

RESULT 3. A basic feasible solution is a minimal feasible solution with total cost \bar{z} if all constants $\bar{c}_{m+1}, \bar{c}_{m+2}, \dots, \bar{c}_n$ are nonnegative, that is, if

$$\bar{c}_j \geq 0 \quad j = m + 1, \dots, n \quad (7.15)$$

The \bar{c}_j are called *reduced costs*.

The proof of this result involves writing the previous equation as

$$f = \bar{f} + \bar{c}_{m+1}x_{m+1} + \dots + \bar{c}_n x_n$$

Because the variables $x_{m+1} \dots x_n$ are presently zero and are constrained to be nonnegative, the only way any one of them can change is for it to become positive. But if $\bar{c}_j \geq 0$ for $j = m + 1, \dots, n$, then increasing any x_j cannot decrease the objective function f because then $\bar{c}_j x_j \geq 0$. Because no feasible change in the nonbasic variables can cause f to decrease, the present solution must be optimal.

The reduced costs also indicate if there are multiple optima. Let all $\bar{c}_j \geq 0$ and let $\bar{c}_k = 0$ for some nonbasic variable x_k . Then, if the constraints allow that variable to be made positive, no change in f results, and there are multiple optima. It is possible, however, that the variable may not be allowed by the constraints to become positive; this may occur in the case of degenerate solutions. We consider the effects of degeneracy later. A corollary to these results is the following:

RESULT 4. A basic feasible solution is the unique minimal feasible solution if $\bar{c}_j > 0$ for all nonbasic variables.

Of course, if some $\bar{c}_j < 0$ then f can be decreased by increasing the corresponding x_j , so the present solution is probably nonoptimal. Thus we must consider means of improving a nonoptimal solution.

Consider the problem of minimizing f , where

$$\begin{aligned} 5x_1 - 4x_2 + 13x_3 - 2x_4 + x_5 &= 20 \\ x_1 - x_2 + 5x_3 - x_4 + x_5 &= 8 \end{aligned} \quad (7.16)$$

$$\begin{aligned} x_1 + 6x_2 - 7x_3 + x_4 + 5x_5 - f &= 0 \\ x_j &\geq 0, \quad j = 1, 2, \dots, 5 \end{aligned} \quad (7.17)$$

We show how the canonical form can be used to improve a nonoptimal basic feasible solution.

Assume that we know that $x_5, x_1, -f$ can be used as basic variables and that the basic solution will be feasible. We can thus reduce system (7.16) to feasible canonical form by pivoting successively on the terms x_5 (first equation) and x_1 (second equation) ($-f$ already appears in the correct way). This yields

$$\begin{aligned} x_5 & - 0.25x_2 + 3x_3 - 0.75x_4 = 5 \\ x_1 & - 0.75x_2 + \textcircled{2x_3} - 0.25x_4 = 3 \\ -f & + 8x_2 - 24x_3 + 5x_4 = -28 \end{aligned} \quad (7.18)$$

The circled term will be explained soon. The basic feasible solution is

$$x_5 = 5, \quad x_1 = 3, \quad x_2 = x_3 = x_4 = 0, \quad f = 28 \quad (7.19)$$

Note that an arbitrary pair of variables does not necessarily yield a basic solution to Equation (7.16) that is feasible. For example, had the variables x_1 and x_2 been chosen as basic variables, the basic solution would have been

$$x_1 = -12, \quad x_2 = -20, \quad x_3 = x_4 = x_5 = 0, \quad f = -132 \quad (7.20)$$

which is not feasible, because x_1 and x_2 are negative.

For the original basic feasible solution, one reduced cost is negative, namely $\bar{c}_3 = -24$. The optimality test of relations (7.15) thus fails. Furthermore, if x_3 is increased from its present value of zero (with all other nonbasic variables remaining zero), f must decrease because, by the third equation of (7.18), f is then related to x_3 by

$$f = 28 - 24x_3 \quad (7.21)$$

How large should x_3 become? It is reasonable to make it as large as possible, because the larger the value of x_3 , the smaller the value of f . The constraints place a limit on the maximum value x_3 can attain, however. Note that, if $x_2 = x_4 = 0$, relations (7.18) state that the basic variables x_1, x_5 are related to x_3 by

$$\begin{aligned} x_5 &= 5 - 3x_3 \\ x_1 &= 3 - 2x_3 \end{aligned} \quad (7.22)$$

Thus as x_3 increases, x_5 and x_1 decrease, and they cannot be allowed to become negative. In fact, as x_3 reaches 1.5, x_1 becomes 0 and as x_3 reaches 1.667, x_5 becomes 0. By that time, however, x_1 is already negative, so the largest value x_3 can attain is

$$x_3 = 1.5 \quad (7.23)$$

Substituting this value into Equations (7.21) and (7.22) yields a new basic feasible solution with lower cost:

$$x_5 = 0.5, x_3 = 1.5, x_1 = x_2 = x_4 = 0, f = -8 \quad (7.24)$$

This solution reduces f from 28 to -8 . The immediate objective is to see if it is optimal. This can be done if the system can be placed into feasible canonical form with $x_5, x_3, -f$ as basic variables. That is, x_3 must replace x_1 as a basic variable. One reason that the simplex method is efficient is that this replacement can be accomplished by doing one pivot transformation.

Previously x_1 had a coefficient of unity in the second equation of (7.18) and zero elsewhere. We now wish x_3 to have this property, and this can be accomplished by pivoting on the term $2x_3$, circled in the second equation of (7.18). This causes x_3 to become basic and x_1 to become nonbasic, as is seen here:

$$\begin{aligned} x_5 & - 1.5x_1 + (0.875x_2) - 0.375x_4 = 0.5 \\ x_3 & + 0.5x_1 - 0.375x_2 - 0.125x_4 = 1.5 \\ -f & + 12x_1 - x_2 + 2x_4 = 8 \end{aligned} \quad (7.25)$$

This gives the basic feasible solution (7.24), as predicted. It also indicates that the present solution although better, is still not optimal, because \bar{c}_2 , the coefficient of x_2 in the f equation, is -1 . Thus we can again obtain a better solution by increasing x_2 while keeping all other nonbasic variables at zero. From Equation (7.25), the current basic variables are then related to x_2 by

$$\begin{aligned} x_5 & = 0.5 - 0.875x_2 \\ x_3 & = 1.5 + 0.375x_2 \\ f & = -8 - x_2 \end{aligned} \quad (7.26)$$

Note that the second equation places no bound on the increase of x_2 , but the first equation restricts x_2 to a maximum of $0.5 / 0.875 = 0.571$, which reduces x_5 to zero. As before, we obtain a new feasible canonical form by pivoting, this time using $0.875x_2$ in the first equation of (7.25) as the pivot term. This yields the system

$$\begin{aligned} x_2 & - 1.714x_1 - 0.429x_4 + 1.142x_5 = 0.571 \\ x_3 & - 0.143x_1 - 10.286x_4 + 0.429x_5 = 1.714 \\ -f & + 10.286x_1 + 1.571x_4 + 1.143x_5 = 8.571 \end{aligned} \quad (7.27)$$

and the basic feasible solution

$$x_2 = 0.571, x_3 = 1.714, x_1 = x_4 = x_5 = 0, f = -8.571 \quad (7.28)$$

Because all reduced costs for the nonbasic variables are positive, this solution is the unique minimal solution of the problem, by the corollary of the previous section. The optimum has been reached in two iterations.

Degeneracy

In the original system (7.18), if the constant on the right-hand side of the second equation had been zero, that is, if the basic feasible solution had been degenerate, then x_1 would have been related to x_3 by

$$x_1 = -2x_3 \quad (7.29)$$

And any positive change in x_3 would have caused x_1 to become negative. Thus x_3 would be forced to remain zero and f could not decrease. We go through the pivot transformation anyway and attain a new form in which the degeneracy may not be limiting. This can easily occur, for if relation (7.29) had been

$$x_1 = 2x_3$$

then x_3 could be made positive.

Unboundedness

If relations (7.26) had been

$$x_5 = 0.5 + 0.875x_2$$

$$x_3 = 1.5 + 0.375x_2$$

$$f = -8 - x_2$$

then x_2 could be made as large as desired without causing x_5 and x_3 to become negative, and f could be made as small as desired. This indicates an unbounded solution. Note that it occurs whenever all coefficients in a column with negative \bar{c}_j are also negative (or zero).

Improving a nonoptimal basic feasible solution in general

Let us now formalize the procedures of the previous section. If at least one $\bar{c}_j < 0$, then, at least if we assume nondegeneracy (all $\bar{b}_i > 0$), it is always possible to construct, by pivoting, another basic feasible solution with lower cost. If more than one $\bar{c}_j < 0$, the variable x_s to be increased can be the one with the most negative \bar{c}_j ; that is, the one whose relative cost factor is

$$\bar{c}_s = \min \bar{c}_j < 0 \quad (7.30)$$

Although this may not lead to the greatest decrease in f (because it may not be possible to increase x_s very far), this is intuitively at least a good rule for choosing the variable to become basic. More sophisticated “pricing” schemes have been developed, however, that perform much better and are included in most modern LP solvers [see Bixby, 1992]. An important recent innovation is the development of steepest edge pricing [see Forrest and Goldfarb (1992)].

Having decided on the variable x_s to become basic, we increase it from zero, holding all other nonbasic variables zero, and observe the effects on the current basic variables. By Equation (7.12), these are related to x_s by

$$\begin{aligned}x_1 &= b_1 - \bar{a}_{1s}x_s \\x_2 &= b_2 - \bar{a}_{2s}x_s \\&\vdots \\x_m &= \bar{b}_m - \bar{a}_{ms}x_s \\f &= \bar{f} + \bar{c}_s x_s, \quad \bar{c}_s < 0\end{aligned}\tag{7.31}$$

Increasing x_s decreases f , and the only factor limiting the decrease is that one of the variables $x_1 \dots x_m$ can become negative. However, if

$$\bar{a}_{is} \leq 0, \quad i = 1, 2, \dots, m\tag{7.32}$$

then x_s can be made as large as desired. Thus we have the following result.

RESULT 5 (UNBOUNDEDNESS). If, in the canonical system for some s , all coefficients \bar{a}_{is} are nonpositive and \bar{c}_s is negative, then a class of feasible solutions can be constructed for which the set of f values has no lower bound.

The class of solutions yielding unbounded f is the set

$$x_i = \bar{b}_i - \bar{a}_{is}x_s, \quad i = 1, \dots, m\tag{7.33}$$

with x_s any positive number and all other $x_i = 0$. If, however, at least one \bar{a}_{is} is positive, then x_s cannot be increased indefinitely because eventually some basic variable becomes first zero, then negative. From Equation (7.31), x_i becomes zero when $\bar{a}_{is} > 0$ and when x_s attains the value

$$x_s = \frac{\bar{b}_i}{\bar{a}_{is}}, \quad \bar{a}_{is} > 0\tag{7.34}$$

The first x_i to become negative is the x_i that requires the smallest x_s to drive it to zero. This value of x_s is the greatest value for x_s permitted by the nonnegativity constraints and is given by

$$x_s^* = \frac{\bar{b}_r}{\bar{a}_{rs}} = \min_{\bar{a}_{is} > 0} \frac{\bar{b}_i}{\bar{a}_{is}}\tag{7.35}$$

The basic variable x_r then becomes nonbasic, to be replaced by x_s . We saw from the example in Equations (7.16)–(7.28) that a new canonical form with x_s replacing x_r as a basic variable is easily obtained by pivoting on the term $\bar{a}_{rs}x_s$. Note that the previous operations may be viewed as simply locating that pivot term. Finding $\bar{c}_s = \min \bar{c}_j < 0$ indicates that the pivot term was in column s , and finding that the minimum of the ratios \bar{b}_i/\bar{a}_{is} for $\bar{a}_{is} > 0$ occurred for $i = r$ indicates that it was in row r .

As seen in the example, if the basic solution is degenerate, then the x_s^* given by Equation (7.35) may be zero. In particular, if some $\bar{b}_i = 0$ and the corresponding $\bar{a}_{is} > 0$ then, by Equation (7.35), $x_s^* = 0$. In this case the pivot operation is still carried out, but f is unchanged.

Iterative procedure

The procedure of the previous section provides a means of going from one basic feasible solution to one whose f is at least equal to the previous f (as can occur, in the degenerate case) or lower, if there is no degeneracy. This procedure is repeated until (1) the optimality test of relations (7.15) is passed or (2) information is provided that the solution is unbounded, leading to the main convergence result.

RESULT 6. Assuming nondegeneracy at each iteration, the simplex algorithm terminates in a finite number of iterations.

Because the number of basic feasible solutions is finite, the algorithm can fail to terminate only if a basic feasible solution is repeated. Such repetition implies that the same value of f is also repeated. Under nondegeneracy, however, each value of f is lower than the previous, so no repetition can occur, and the algorithm is finite.

Degenerate case

If, at some iteration, the basic feasible solution is degenerate, the possibility exists that f can remain constant for some number of subsequent iterations. It is then possible for a given set of basic variables to be repeated. An endless loop is then set up, the optimum is never attained, and the simplex algorithm is said to have cycled. Examples of cycling have been constructed [see Dantzig (1998), Chapter 10].

Some procedures are guaranteed to avoid cycling (Dantzig, 1998). Modern LP solvers contain very effective antidegeneracy strategies, although most are not guaranteed to avoid cycling. In practice, almost all LPs have degenerate optimal solutions. A high degree of degeneracy (i.e., a high percentage of basic variables at bounds) can slow the simplex method down considerably. Fortunately, an alternative class of LP algorithms, called *barrier methods*, are not affected by degeneracy. We discuss these briefly later in the chapter.

Two phases of the simplex method

The simplex algorithm requires a basic feasible solution as a starting point. Such a starting point is not always easy to find and, in fact, none exists if the constraints are inconsistent. Phase 1 of the simplex method finds an initial basic feasible solution or yields the information that none exists. Phase 2 then proceeds from this starting

point to an optimal solution or yields the information that the solution is unbounded. Both phases use the simplex algorithm of the previous section.

Phase 1. Phase 1 starts with some initial basis \mathbf{B} and an initial basic (possibly infeasible) solution $(\mathbf{x}_B, \mathbf{x}_N)$ satisfying

$$\mathbf{B}\mathbf{x}_B + \mathbf{N}\mathbf{x}_N = \mathbf{b} \quad (7.36)$$

In the previous expression, all components of \mathbf{x}_N are at bounds and \mathbf{N} is the corresponding matrix of coefficients for \mathbf{x}_N . Because \mathbf{B} is nonsingular

$$\mathbf{x}_B = \mathbf{B}^{-1}(\mathbf{b} - \mathbf{N}\mathbf{x}_N) \quad (7.37)$$

If \mathbf{x}_B is between its bounds, the basic solution is feasible and we begin phase 2, which optimizes the true objective. Otherwise, some components of \mathbf{x}_B violate their bounds. Let L and U be the sets of indices of basic variables that violate their bounds, that is

$$x_j < l_j, \quad j \in L \quad (7.38)$$

and

$$x_j > u_j, \quad j \in U \quad (7.39)$$

Phase 1 minimizes the following linear objective function, the sum of infeasibilities, sinf ;

$$\text{sinf} = \sum_{j \in L} (l_j - x_j) + \sum_{j \in U} (x_j - u_j) \quad (7.40)$$

Note that each term is positive, and that $\text{sinf} = 0$ if and only if the basic solution is feasible. When minimizing sinf , the standard simplex algorithm is applied, but the rules for choosing the pivot row described earlier must be changed, because some basic variables are now infeasible. During this process, infeasible basic variables can satisfy their bounds and feasible ones can violate their bounds, so the index sets L and U (and hence the function sinf) can change at any iteration. If the simplex optimality test is met and $\text{sinf} > 0$, then the LP is infeasible. Otherwise, when $\text{sinf} = 0$, phase 2 begins using the simplex method discussed earlier.

The initial basis

Often a good initial basis is known. Once an LP model is constructed and validated, it is common to do several series of case studies. In each case study, a set of LP data elements (cost or right-hand side components, bounds, or matrix elements a_{ij}) are assigned a sequence of closely related sets of values. For example, one may vary several costs through a range of values or equipment capacities or customer demands (both of the last two are right-hand sides or bounds). If there are several sets of parameter values, after the first set is solved, the optimal basis is stored and used as the initial basis for the LP problem that uses the second set, and so on. This usually sharply reduces computation time compared with a cold start, where no good initial basis is known. In fact, the simplex method's ability to warm start effectively is one of its major advantages over barrier methods (discussed later).

EXAMPLE 7.3 ITERATIVE SOLUTION OF AN LP PROBLEM

Consider first the problem illustrated geometrically in Figure 7.1 given in relations (7.1), that is

$$\begin{aligned} \text{Maximize: } f &= x_1 + 3x_2 \\ \text{Subject to: } -x_1 + x_2 + x_3 &= 1 \\ x_1 + x_2 + x_4 &= 2 \\ x_i &\geq 0, \quad i = 1, \dots, 4 \end{aligned} \quad (a)$$

where x_3, x_4 are slack variables. Solve for the maximum using the simplex method.

Solution. Here no phase 1 is needed because an initial basic feasible solution is obvious. To apply directly the results of the previous sections, we rephrase the problem as

$$\text{Minimize: } -x_1 - 3x_2$$

subject to Equation (a). The initial feasible canonical form is

$$\begin{aligned} -x_1 + \textcircled{x_2} + x_3 &= 1 \\ x_1 + x_2 + x_4 &= 2 \\ -x_1 - 3x_2 - f &= 0 \end{aligned} \quad (b)$$

The initial basic feasible solution is

$$x_1 = x_2 = 0, \quad x_3 = 1, \quad x_4 = 2, \quad f = 0 \quad (c)$$

This corresponds to vertex (2) of Figure 7.1.

Iteration 1. Because $\bar{c}_2 = \min(\bar{c}_1, \bar{c}_2) = -3 < 0$, x_2 becomes basic. To see which variable becomes nonbasic, we compute the ratios b_i/a_{i2} ; for all i such that $\bar{a}_{i2} > 0$. This gives

$$\frac{\bar{b}_1}{\bar{a}_{12}} = \frac{1}{1} = 1, \quad \frac{\bar{b}_2}{\bar{a}_{22}} = \frac{2}{1} = 2$$

The minimum of these is \bar{b}_1/\bar{a}_{12} ; thus the basic variable with unity coefficient in row 1, x_3 , leaves the basis. The pivot term is $a_{12}x_2$ that is, the x_2 term circled in Equation (b). Pivoting on this term yields

$$\begin{aligned} -x_1 + x_2 + x_3 &= 1 \\ \textcircled{2x_1} - x_3 + x_4 &= 1 \\ -4x_1 + 3x_3 - f &= 3 \end{aligned} \quad (d)$$

Iteration 2. The new basic feasible solution is

$$x_1 = x_3 = 0, \quad x_2 = x_4 = 1, \quad f = -3$$

Note that f is reduced. The solution corresponds to vertex (3) of Figure 7.1. Because $\bar{c}_1 = -4 = \min_j \bar{c}_j$, x_1 becomes basic. The only ratio \bar{b}_i/\bar{a}_{i1} having $\bar{a}_{i1} > 0$ is that for $i = 2$; thus x_4 becomes nonbasic and the circled pivot term is $\bar{a}_{21}x_1 = 2x_1$. Pivoting yields

$$\begin{aligned} x_2 + 0.5x_3 + 0.5x_4 &= 1.5 \\ x_1 - 0.5x_3 + 0.5x_4 &= 0.5 \\ x_3 + 2x_4 - f &= 5 \end{aligned} \tag{e}$$

with basic feasible solution

$$x_1 = 0.5, x_2 = 1.5, x_3 = x_4 = 0, f = -5$$

which corresponds to vertex (4) of Figure 7.1. This is optimal, since all $\bar{c}_j > 0$. The path taken by the method is vertices (2), (3), (4).

7.4 BARRIER METHODS

Barrier methods for linear programming were first proposed in the 1980s and are now included in most commercial LP software systems. Their underlying principles and the way they operate are very different from the simplex method. They generate a sequence of points that may not satisfy all the constraints until the method converges and none of the points need be extreme points. This allows them to cut across the feasible region rather than moving from one extreme point to another, as the simplex method does. Hence they usually take far fewer iterations than the simplex method, but each iteration takes more time. See Martin (1999), Vanderbei (1999), or Wright (1999) for complete explanations. Current implementations of barrier methods are competitive with the best simplex codes, are often faster on very large problems, and often do very well in problems where the simplex method is slowed by degeneracy.

7.5 SENSITIVITY ANALYSIS

In addition to providing optimal x values, both simplex and barrier solvers provide values of dual variables or Lagrange multipliers for each constraint. We discuss Lagrange multipliers at some length in Chapter 8, and the conclusions reached there, valid for nonlinear problems, must hold for linear programs as well. In Chapter 8 we show that the dual variable for a constraint is equal to the derivative of the optimal objective value with respect to the constraint limit or right-hand side. We illustrate this with examples in Section 7.8.

7.6 LINEAR MIXED INTEGER PROGRAMS

A mixed integer linear program (MILP) is an LP in which one or more of the decision variables must be integers. A common subset of MILPs are binary, in which the integer variables can be either 0 or 1, indicating that something is either done or not done. For example, the binary variable $x_j = 1(0)$ can mean that a facility is (is not) placed at location j , or project j is (is not) selected. For such yes–no variables, fractional values have no significance. Almost all LP solvers now include the capability to solve MILPs, and this dramatically increases their usefulness. The computational difficulty of solving MILPs is determined mainly by the number of integer variables, and only in a secondary way by the number of continuous variables or constraints. Currently the best MILP solvers can handle hundreds of integer variables in reasonable time, sometimes more, depending on the problem structure and data. We discuss MILP's further in Chapter 9 [see also Martin (1999) and Wolsey (1998)].

7.7 LP SOFTWARE

LP software includes two related but fundamentally different kinds of programs. The first is solver software, which takes data specifying an LP or MILP as input, solves it, and returns the results. Solver software may contain one or more algorithms (simplex and interior point LP solvers and branch-and-bound methods for MILPs, which call an LP solver many times). Some LP solvers also include facilities for solving some types of nonlinear problems, usually quadratic programming problems (quadratic objective function, linear constraints; see Section 8.3), or separable nonlinear problems, in which the objective or some constraint functions are a sum of nonlinear functions, each of a single variable, such as

$$f(x) = x_1^2 + e^{x_2} + 7/x_3 - 2x_4$$

Modeling systems

A second feature of LP programs is the inclusion of modeling systems, which provide an environment for formulating, solving, reporting on, analyzing, and managing LP and MILP models. Modeling systems have links to several LP, MILP, and NLP solvers and allow users to change solvers by changing a single statement. Modeling systems are all designed around a language for formulating optimization models, and most are capable of formulating and solving both linear and nonlinear problems. *Algebraic modeling systems* represent optimization problems using algebraic notation and a powerful indexing capability. This allows sets of similar constraints to be represented by a single modeling statement, regardless of the number of constraints in the set. For more information on algebraic modeling languages, see Section 8.9.3.

Another type of widely used modeling system is the *spreadsheet solver*. Microsoft Excel contains a module called the Excel Solver, which allows the user to enter the decision variables, constraints, and objective of an optimization problem into the cells of a spreadsheet and then invoke an LP, MILP, or NLP solver. Other spreadsheets contain similar solvers. For examples using the Excel Solver, see Section 7.8, and Chapters 8 and 9.

The power of linear programming solvers

Modern LP solvers can solve very large LPs very quickly and reliably on a PC or workstation. LP size is measured by several parameters: (1) the number of variables n , (2) the number of constraints m , and (3) the number of nonzero entries nz in the constraint matrix A . The best measure is the number of nonzero elements nz because it directly determines the required storage and has a greater effect on computation time than n or m . For almost all LPs encountered in practice, nz is much less than mn , because each constraint involves only a few of the variables x . The *problem density* $100(nz/mn)$ is usually less than 1%, and it almost always decreases as m and n increase. Problems with small densities are called *sparse*, and real world LPs are always sparse. Roughly speaking, a problem with under 1000 nonzeros is small, between 1000 and 50,000 is medium-size, and over 50,000 is large. A small problem probably has m and n in the hundreds, a medium-size problem in the low to mid thousands, and a large problem above 10,000.

Currently, a good LP solver running on a fast (> 500 MHz) PC with substantial memory, solves a small LP in less than a second, a medium-size LP in minutes to tens of minutes, and a large LP in an hour or so. These codes hardly ever fail, even if the LP is badly formulated or scaled. They include preprocessing procedures that detect and remove redundant constraints, fixed variables, variables that must be at bounds in any optimal solution, and so on. Preprocessors produce an equivalent LP, usually of reduced size. A postprocessor then determines values of any removed variables and Lagrange multipliers for removed constraints. Automatic scaling of variables and constraints is also an option. Armed with such tools, an analyst can solve virtually any LP that can be formulated.

Solving MILPs is much harder. Focusing on MILPs with only binary variables, problems with under 20 binary variables are small, 20 to 100 is medium-size, and over 100 is large. Large MILPs may require many hours to solve, but the time depends greatly on the problem structure and the availability of a good starting point. We discuss MILP and MINLP formulations in Chapter 9.

Imbedded Linear Programming solvers

In addition to their use as stand-alone systems, LPs are often included within larger systems intended for decision support. In this role, the LP solver is usually hidden from the user, who sees only a set of critical problem input parameters and a set of suitably formatted solution reports. Many such systems are available for supply chain management—for example, planning raw material acquisitions and deliveries, production and inventories, and product distribution. In fact, the process industries—oil, chemicals, pharmaceuticals—have been among the earliest users. Almost every refinery in the developed world plans production using linear programming.

When embedded in decision support systems (usually in a Windows environment), LP solvers typically receive input data from a program written in C or Visual Basic and are often in the form of dynamic link libraries (DLLs). Most of today's LP solvers are available as DLLs.

Available Linear Programming software

Many LP software vendors advertise in the monthly journal *OR/MS Today*, published by INFORMS. For a survey of LP software, see Fourer (1997, 1999) in that journal. All vendors now have Websites, and the following table provides a list of LP software packages along with their Web addresses.

Company name	Solver name	Web addresses/E-mail address
CPLEX Division of ILOG	CPLEX	www.cplex.com
IBM	Optimization Software Library (OSL)	www.research.ibm.com/osl/
LINDO Systems Inc.	LINDO	www.lindo.com
Dash Associates	XPRESS-MP	www.dashopt.com
Sunset Software Technology	AXA	Sunsetw@ix.netcom.com
Advanced Mathematical Software	LAMPS	info@amssoft.demon.co.uk

7.8 A TRANSPORTATION PROBLEM USING THE EXCEL SOLVER SPREADSHEET FORMULATION

Figure 7.3 displays a Microsoft Excel spreadsheet containing the formulas and data for an LP transportation problem. This spreadsheet is one of six optimization examples included with Microsoft Excel '97. With a standard installation of Microsoft Office, the Excel workbook containing all six examples is in the file

MicrosoftOffice/office/examples/solver/solvsamp.xls

We encourage the reader to start Excel on his or her computer, find and open this file, and examine and solve this spreadsheet as the rest of this section is read. The 15 decision variables are the number of units of a single product to ship from three plants to five warehouses. Initial values of these variables (all ones) are in the range C8:G10. The constraints are (1) the amount shipped from each plant cannot exceed the available supply, given in range B16:B18, (2) the amount shipped to each warehouse must meet or exceed demand there, given in range C14:G14, and (3) all amounts shipped must be nonnegative. Cells C16:G18 contain the per unit costs of shipping the product along each of the 15 possible routes. The total cost of shipping into each warehouse is in the range C20:G20, computed by multiplying the amounts shipped by their per unit costs and summing. Total shipping cost in cell B20 is to be minimized. Before reading further, attempt to find an optimal solution to this problem by trying your own choices for the decision variables.

	A	B	C	D	E	F	G	H	I	J	K	
1	Example 2: Transportation Problem.											
2	Minimize the costs of shipping goods from production plants to warehouses near metropolitan demand											
3	centers, while not exceeding the supply available from each plant and meeting the demand from each											
4	metropolitan area.											
6			Number to ship from plant <i>x</i> to warehouse <i>y</i> (at intersection):									
7	Plants:	Total	San Fran	Denver	Chicago	Dallas	New York					
8	S. Carolina	5	1	1	1	1	1					
9	Tennessee	5	1	1	1	1	1					
10	Arizona	5	1	1	1	1	1					
11			-	-	-	-	-					
12	Totals:		3	3	3	3	3					
13												
14	Demands by Whse →		180	80	200	160	220					
15	Plants:	Supply	Shipping costs from plant <i>x</i> to warehouse <i>y</i> (at intersection):									
16	S. Carolina	310	10	8	6	5	4					
17	Tennessee	260	6	5	4	3	6					
18	Arizona	280	3	4	5	5	9					
19												
20	Shipping:	\$83	\$19	\$17	\$15	\$13	\$19					
21												
22	The problem presented in this model involves the shipment of goods from three plants to five regional											
23	warehouses. Goods can be shipped from any plant to any warehouse, but it obviously costs more to											
24	ship goods over long distances than over short distances. The problem is to determine the amounts											
25	to ship from each plant to each warehouse at minimum shipping cost in order to meet the regional											
26	demand, while not exceeding the plant supplies.											
27												
28	Problem Specifications											
29												
30	Target cell		B20									
31												
32	Changing cells		C8:G10									
33												
35	Constraints		B8:B10<=B16:B18									
36												
38			C12:G12>=C14:G14									
39												
41			C8:G10>=0									
42												
43												
44	You can solve this problem faster by selecting the Assume linear model check box in the Solver											
45	Options dialog box before clicking Solve . A problem of this type has an optimum solution at which											
46	amounts to ship are integers, if all of the supply and demand constraints are integers.											
47												

FIGURE 7.3

A transportation problem in a Microsoft Excel spreadsheet format. Permission by Microsoft.

Algebraic formulation

Let x_{ij} be the number of units of the product shipped from plant i to warehouse j . Then the supply constraints are

$$\sum_{j=1}^5 x_{ij} \leq \text{avail}_i, \quad i = 1, 2, 3 \quad (7.41)$$

The demand constraints are

$$\sum_{i=1}^3 x_{ij} \geq \text{demand}_j, \quad j = 1, \dots, 5 \quad (7.42)$$

and the nonnegativities:

$$x_{ij} \geq 0, \quad \text{all } i, j \quad (7.43)$$

The objective is to minimize

$$\text{Cost} = \sum_{j=1}^5 \sum_{i=1}^3 c_{ij} x_{ij} \quad (7.44)$$

Solver parameters dialog

To define this problem for the Excel Solver, the cells containing the decision variables, the constraints, and the objective must be specified. This is done by choosing the Solver command from the Tools menu, which causes the Solver parameters dialog shown in Figure 7.4 to appear. The “Target Cell” is the cell containing the objective function. Clicking the “Help” button explains all the steps needed to enter the “changing” (i.e., decision) variables and the constraints. We encourage you to “Reset all,” and fill in this dialog from scratch.

Solver options dialog

Selecting the “Options” button in the Solver Parameters dialog brings up the Solver Options dialog box shown in Figure 7.5. The current Solver version does not determine automatically if the problem is linear or nonlinear. To inform Solver that

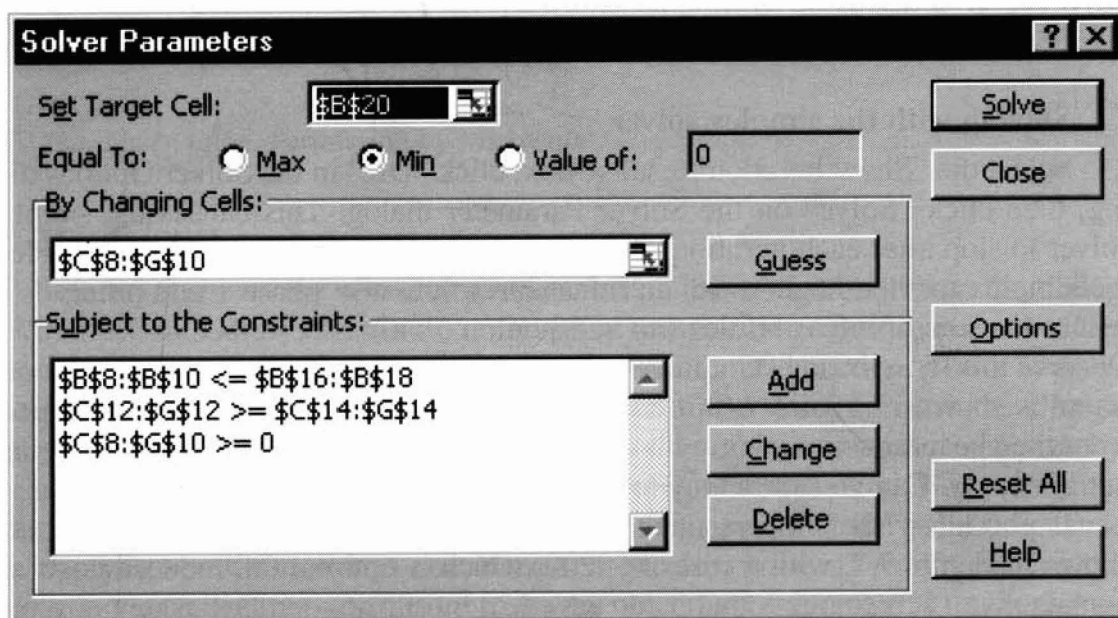


FIGURE 7.4

Solver parameters dialog box. Permission by Microsoft.

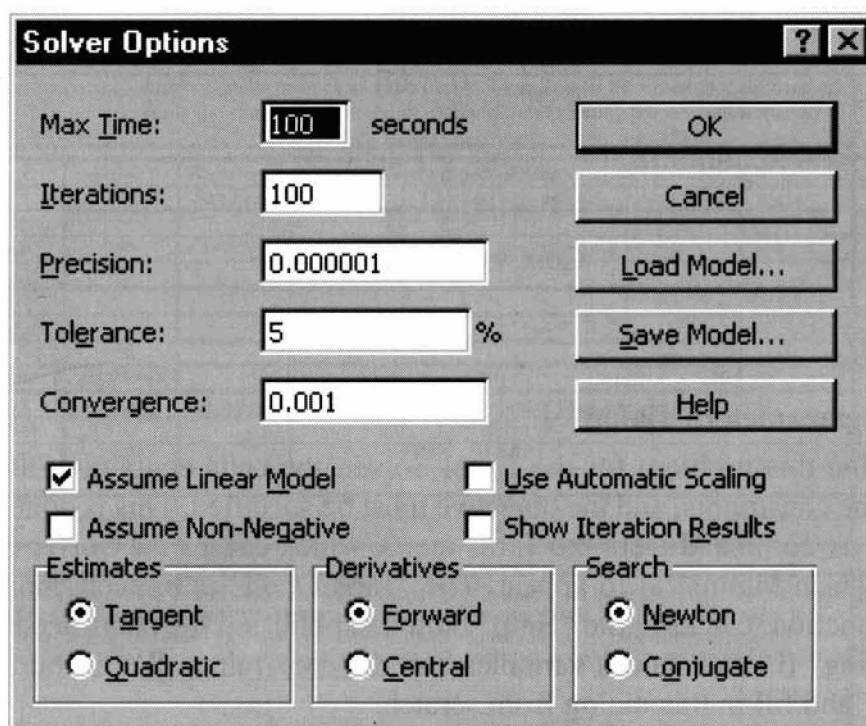


FIGURE 7.5
Solver options dialog box. Permission by Microsoft.

the problem is an LP, select the “Assume Linear Model” box. This causes the simplex solver to be used. It is both faster and more accurate for LPs than the generalized reduced gradient (GRG) nonlinear solver, which is the default choice. The GRG solver is discussed in Chapter 8.

Solving with the simplex solver

Select the “Show Iteration Results” box, click “OK” in the Solver Options dialog, then click “Solve” on the Solver Parameter dialog. This causes the simplex solver to stop after each iteration. Because an initial feasible basis is not provided, the simplex method begins with an infeasible solution in phase 1 and proceeds to reduce the sum of infeasibilities sinf in Equation (7.40) as described in Section 7.3. Observe this by selecting “Continue” after each iteration. The first feasible solution found is shown in Figure 7.6. It has a cost of \$3210, with most shipments made from the cheapest source, but with other sources used when the cheapest one runs out of supply. Can you see a way to improve this solution?

If you allow the simplex method to continue, it finds the improved solution shown in Figure 7.7, with a cost of \$3200, which is optimal (all reduced costs are nonnegative). It recognizes that it can save \$20 by shifting ten Dallas units from S. Carolina to Tennessee, if it frees up ten units of supply at Tennessee by supplying Chicago from Arizona (which costs only \$10 more). Supplies at Arizona and Tennessee are completely used, but South Carolina has ten units of excess supply.

	A	B	C	D	E	F	G	H	I	J	K
1	Example 2: Transportation Problem.										
2	Minimize the costs of shipping goods from production plants to warehouses near metropolitan demand										
3	centers, while not exceeding the supply available from each plant and meeting the demand from each										
4	metropolitan area.										
6	<i>Number to ship from plant x to warehouse y (at intersection):</i>										
7	<i>Plants:</i>	<i>Total</i>	<i>San Fran</i>	<i>Denver</i>	<i>Chicago</i>	<i>Dallas</i>	<i>New York</i>				
8	S. Carolina	310	0	0	0	90	220				
9	Tennessee	260	0	0	190	70	0				
10	Arizona	270	180	80	10	0	0				
11			-	-	-	-	-				
12	Totals:		180	80	200	160	220				
13											
14	<i>Demands by Whse →</i>		180	80	200	160	220				
15	<i>Plants:</i>	<i>Supply</i>	<i>Shipping costs from plant x to warehouse y (at intersection):</i>								
16	S. Carolina	310	10	8	6	5	4				
17	Tennessee	260	6	5	4	3	6				
18	Arizona	280	3	4	5	5	9				
19											
20	<i>Shipping:</i>	\$3,210	\$540	\$320	\$810	\$660	\$880				
21											
22	The problem presented in this model involves the shipment of goods from three plants to five regional										
23	warehouses. Goods can be shipped from any plant to any warehouse, but it obviously costs more to										
24	ship goods over long distances than over short distances. The problem is to determine the amounts										
25	to ship from each plant to each warehouse at minimum shipping cost in order to meet the regional										
26	demand, while not exceeding the plant supplies.										
27											
28	Problem Specifications										
29											
30	Target cell	B20		Goal is to minimize total shipping cost.							
31											
32	Changing cells	C8:G10		Amount to ship from each plant to each							
33	warehouse.										
34											
35	Constraints	B8:B10<=B16:B18		Total shipped must be less than or equal to							
36	supply at plant.										
37											
38		C12:G12>=C14:G14		Totals shipped to warehouses must be greater							
39	than or equal to demand at warehouses.										
40											
41		C8:G10>=0		Number to ship must be greater than or equal							
42	to 0.										
43											
44	You can solve this problem faster by selecting the Assume linear model check box in the Solver										
45	Options dialog box before clicking Solve . A problem of this type has an optimum solution at which										
46	amounts to ship are integers, if all of the supply and demand constraints are integers.										
47											

FIGURE 7.6
First feasible solution. Permission by Microsoft.

The sensitivity report

Figure 7.8 shows the sensitivity report, which can be selected from the dialog box that appears when the solution algorithm finishes. The most important information is the “Shadow Price” column in the “constraints” section. These shadow prices (also called dual variables or Lagrange multipliers) are equal to the change in the optimal objective value if the right-hand side of the constraint increases by one unit, with all other right-hand side values remaining the same. Hence the first three multipliers show the effect of increasing the supplies at the plants. Because the supply in South Carolina is not all used, its shadow price is zero. Increasing the supply in Tennessee by one unit improves the objective by 2, twice as much as Arizona. To verify this, increase the Tennessee supply to 261, resolve, and observe that the new objective value is \$3198. The last five shadow prices show the effects of increasing the demands. The “Allowable Increase” is the amount the right-hand

	A	B	C	D	E	F	G	H	I	J	K	
1	Example 2: Transportation Problem.											
2	Minimize the costs of shipping goods from production plants to warehouses near metropolitan demand											
3	centers, while not exceeding the supply available from each plant and meeting the demand from each											
4	metropolitan area.											
6			<i>Number to ship from plant x to warehouse y (at intersection):</i>									
7	<i>Plants:</i>	<i>Total</i>	<i>San Fran</i>	<i>Denver</i>	<i>Chicago</i>	<i>Dallas</i>	<i>New York</i>					
8	S. Carolina	300	0	0	0	80	220					
9	Tennessee	260	0	0	180	80	0					
10	Arizona	280	180	80	20	0	0					
11			–	–	–	–	–					
12	Totals:		180	80	200	160	220					
13												
14	<i>Demands by Whse →</i>		180	80	200	160	220					
15	<i>Plants:</i>	<i>Supply</i>	<i>Shipping costs from plant x to warehouse y (at intersection):</i>									
16	S. Carolina	310	10	8	6	5	4					
17	Tennessee	260	6	5	4	3	6					
18	Arizona	280	3	4	5	5	9					
19												
20	<i>Shipping:</i>	\$3,200	\$540	\$320	\$820	\$640	\$880					
21												
22	The problem presented in this model involves the shipment of goods from three plants to five regional											
23	warehouses. Goods can be shipped from any plant to any warehouse, but it obviously costs more to											
24	ship goods over long distances than over short distances. The problem is to determine the amounts											
25	to ship from each plant to each warehouse at minimum shipping cost in order to meet the regional											
26	demand, while not exceeding the plant supplies.											
27												
28	Problem Specifications											
30	Target cell		B20					Goal is to minimize total shipping cost.				
32	Changing cells		C8:G10					Amount to ship from each plant to each warehouse.				
35	Constraints		B8:B10<=B16:B18					Total shipped must be less than or equal to supply at plant.				
38			C12:G12>=C14:G14					Totals shipped to warehouses must be greater than or equal to demand at warehouses.				
41			C8:G10>=0					Number to ship must be greater than or equal to 0.				
44	You can solve this problem faster by selecting the Assume linear model check box in the Solver											
45	Options dialog box before clicking Solve . A problem of this type has an optimum solution at which											
46	amounts to ship are integers, if all of the supply and demand constraints are integers.											

FIGURE 7.7
Optimal solution. Permission by Microsoft.

side can increase before the shadow price changes, and similarly for the “Allowable Decrease.” Beyond these ranges, some shipment that is now zero becomes positive while some positive one becomes zero. Try right-hand side changes within and slightly beyond one of the ranges to verify this.

The “Adjustable Cells” section contains sensitivity information on changes in the objective coefficients. The reduced costs are the qualities \bar{c}_j discussed in Section 7.3. These are all nonnegative, as they must be in an optimal solution—see result 3. Note that the \bar{c}_j for the South Carolina–Chicago shipment is zero, indicating that this problem has multiple optima (because this optimal solution is nondegenerate, i.e., all basic variables are positive). The following table shows a set of shipping unit amounts that yields no net cost change.

Adjustable Cells

Cell	Name	Final Value	Reduced Cost	Objective Coefficient	Allowable Increase	Allowable Decrease
\$C\$8	S. Carolina San Fran	0	6	10	1E+30	6
\$D\$8	S. Carolina Denver	0	3	8	1E+30	3
\$E\$8	S. Carolina Chicago	0	0	6	1E+30	0
\$F\$8	S. Carolina Dallas	80	0	5	0	1
\$G\$8	S. Carolina New York	220	0	4	4	4
\$C\$9	Tennessee San Fran	0	4	6	1E+30	4
\$D\$9	Tennessee Denver	0	2	5	1E+30	2
\$E\$9	Tennessee Chicago	180	0	4	0	1
\$F\$9	Tennessee Dallas	80	0	3	1	0
\$G\$9	Tennessee New York	0	4	6	1E+30	4
\$C\$10	Arizona San Fran	180	0	3	4	4
\$D\$10	Arizona Denver	80	0	4	2	5
\$E\$10	Arizona Chicago	20	0	5	1	2
\$F\$10	Arizona Dallas	0	1	5	1E+30	1
\$G\$10	Arizona New York	0	6	9	1E+30	6

Constraints

Cell	Name	Final Value	Shadow Price	Constraint R.H. Side	Allowable Increase	Allowable Decrease
\$B\$8	S. Carolina Supply	300	0	310	1E+30	10
\$B\$9	Tennessee Supply	260	-2	260	80	10
\$B\$10	Arizona Supply	280	-1	280	80	10
\$C\$12	San Fran Demand	180	4	180	10	80
\$D\$12	Denver Demand	80	5	80	10	80
\$E\$12	Chicago Demand	200	6	200	10	80
\$F\$12	Dallas Demand	160	5	160	10	80
\$G\$12	New York Demand	220	4	220	10	220

FIGURE 7.8
Sensitivity report.

Shipment	Change	Cost change
South Carolina–Chicago	+1	+6
Tennessee–Chicago	-1	-4
Tennessee–Dallas	+1	+3
S. Carolina–Dallas	-1	-5
Total		0

These changes leave the amounts shipped out from the plants and into the warehouses unchanged.

7.9 NETWORK FLOW AND ASSIGNMENT PROBLEMS

This transportation problem is an example of an important class of LPs called *network flow problems*: Find a set of values for the flow of a single commodity on the arcs of a graph (or network) that satisfies both flow conservation constraints at each node (i.e., flow in equals flow out) and upper and lower limits on each flow, and maximize or minimize a linear objective (say, total cost). There are specified supplies of the commodity at some nodes and demands at others. Such problems have the important special property that, if all supplies, demands, and flow bounds are integers, then an optimal solution exists in which all flows are integers. In addition, special versions of the simplex method have been developed to solve network flow problems with hundreds of thousands of nodes and arcs very quickly, at least ten times faster than a general LP of comparable size. See Glover et al. (1992) for further information.

The integer solution property is particularly important in assignment problems. These are transportation problems (like the problem just described) with n supply nodes and n demand nodes, where each supply and demand is equal to 1.0, and all constraints are equalities. Then the model in Equations (7.41) through (7.44) has the following interpretation: Each supply node corresponds to a “job,” and each demand node to a “person.” The problem is to assign each “job” to a “person” so that some measure of benefit or cost is optimized. The variables x_{ij} are 1 if “job” i is assigned to “person” j , and zero otherwise.

As an example, suppose we want to assign streams to heat exchangers and the cost (in some measure) of doing so is listed in the following matrix:

		Exchanger number			
		1	2	3	4
<i>Stream</i>	<i>A</i>	94	1	54	68
	<i>B</i>	74	10	88	82
	<i>C</i>	73	88	8	76
	<i>D</i>	11	74	81	21

Each element in the matrix represents the cost of transferring stream i to exchanger j . How can the cost be minimized if each stream goes to only one exchanger?

First let us write the problem statement. The total number of streams n is 4. Let c_{ij} be an element of the cost matrix, which is the cost of assigning stream i to exchanger j . Then we have the following assignment problem:

$$\text{Minimize: } f(\mathbf{x}) = \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij}$$

$$\sum_{i=1}^n x_{ij} = 1 \quad j = 1, \dots, n \quad (7.45)$$

$$\sum_{j=1}^n x_{ij} = 1 \quad i = l, \dots, n$$

$$x_{ij} \geq 0 \quad i, j = l, \dots, n$$

The constraints (7.46) ensure that each stream is assigned to some exchanger, and Equation (7.45) ensures that each exchanger is assigned one stream. Because the supplies and demands are integers, this problem has an optimal integer solution, with each x_{ij} equal to 0 or 1. The reader is invited to solve this problem using the Excel Solver (or any other LP solver) and find an optimal assignment.

REFERENCES

- Bixby, R. E. "Implementing the Simplex Method: The Initial Basis." *ORSA J Comput* **4**(3): 267–284 (1992).
- Dantzig, G. B. *Linear Programming and Extensions*. Princeton University Press, Princeton, NJ (1998).
- Forrest, J. J.; and D. Goldfarb. "Steepest-edge Simplex Algorithms for Linear Programming." *Math Prog* **57**: 341–374 (1992).
- Fourer, R. "Linear Programming." *OR/MS Today* **24**: 54–67 (1997).
- Fourer, R. "Software for Optimization." *OR/MS Today* **26**: 40–44 (1999).
- Glover, F.; D. Klingman; and N. Phillips. *Network Models in Optimization and Their Applications in Practice*. Wiley, New York (1992).
- Martin, R. K. *Large Scale Linear and Integer Optimization*. Kluwer Academic Publishers, Norwell, MA (1999).
- Vanderbei, R. J. *Linear Programming Foundations and Extensions*. Kluwer Academic Publishers, Norwell, MA (1999).
- Wolsey, L. A. *Integer Programming*. Wiley, New York (1998).
- Wright, S. J. *Primal-Dual Interior-Point Methods*. SIAM, Philadelphia, PA (1999).

SUPPLEMENTARY REFERENCES

- Darst, R. B. *Introduction to Linear Programming*. Marcel Dekker, New York (1990).
- Gill, P. E.; W. Murray; M. A. Saunders; J. A. Tomlin; and M. H. Wright. "On Projected Newton Barrier Methods for Linear Programming and an Equivalence to Karmarkar's Projective Method." *Math Program* **36**: 183–191 (1986).
- Gass, S. I. *An Illustrated Guide to Linear Programming*. Dover, Mineola, NY (1990).
- Johnson, J. D.; and C. Q. Williamson. "In-Line Gasoline Blending at Suntide Refinery." *IEEE Trans. Industry and General Applications*, **159**: 167–179 (March/April, 1967).
- Karmarkar, N. "A New Polynomial-Time Algorithm for Linear Programming." *Combinatorica* **4**: 373–383 (1984).
- Murtagh, B. A. *Advanced Linear Programming*. McGraw-Hill, New York (1983).
- Murty, K. G. *Linear Programming*. Wiley, New York (1983).
- Perry C.; and R. Crellin. "The Precise Meaning of a Shadow Price." *Interfaces* **12**: 61–68 (1982).

- Shamir, R. "The Efficiency of the Simplex Method: A Survey." *Man. Sci.* **33**: 301–310 (1987).
- Schrage, L. *Optimization Modeling with LINDO*. Duxbury Press, Pacific Grove, CA (1997).
- Schrijver, A. *Theory of Linear and Integer Programming*. Wiley, New York (1986).
- Snee, R. D. "Developing Blending Models for Gasoline and Other Mixtures." *Technometrics* **23**: 119–127 (1981).
- Sourander, M. L.; M. Kolari; J. C. Cugini; J. B. Poje; and D. C. White. "Control and Optimization of Olefin-Cracking Heaters." *Hydrocarbon Process.* pp. 63–68 (June, 1984).
- Ye, Y. *Interior Point Algorithms: Theory and Analysis*. Wiley, New York (1997).

PROBLEMS

- 7.1** A refinery has available two crude oils that have the yields shown in the following table. Because of equipment and storage limitations, production of gasoline, kerosene, and fuel oil must be limited as also shown in this table. There are no plant limitations on the production of other products such as gas oils.

The profit on processing crude #1 is \$1.00/bbl and on crude #2 it is \$0.70/bbl. Find the approximate optimum daily feed rates of the two crudes to this plant via a graphical method.

	Volume percent yields		Maximum allowable product rate (bbl/day)
	Crude #1	Crude #2	
Gasoline	70	31	6,000
Kerosene	6	9	2,400
Fuel oil	24	60	12,000

- 7.2** A confectioner manufactures two kinds of candy bars: Ergies (packed with energy for the kiddies) and Nergies (the "lo-cal" nugget for weight watchers without willpower). Ergies sell at a profit of 50¢ per box, and Nergies have a profit of 60¢ per box. The candy is processed in three main operations: blending, cooking, and packaging. The following table records the average time in minutes required by each box of candy, for each of the three activities.

	Blending	Cooking	Packing
Ergies	1	5	3
Nergies	2	4	1

During each production run, the blending equipment is available for a maximum of 14 machine hours, the cooking equipment for at most 40 machine hours, and the packaging equipment for at most 15 machine hours. If each machine can be allocated to the making of either type of candy at all times that it is available for production, determine how many boxes of each kind of candy the confectioner should make to realize the maximum profit. Use a graphical technique for the two variables.

- 7.3 Feed to three units is split into three streams: F_A , F_B , and F_C . Two products are produced: P_1 and P_2 (see following figure), and the yield in weight percent by unit is

Yield (weight %)	Unit A	Unit B	Unit C
P_1	40	30	50
P_2	60	70	50

Each stream has values in \$/lb as follows:

Stream	F	P_1	P_2
Value (\$/lb)	.40	.60	.30

Because of capacity limitations, certain constraints exist in the stream flows:

1. The total input feed must not exceed 10,000 lb/day.
2. The feed to each of the units A, B, and C must not exceed 5000 lb/day.
3. No more than 4000 lb/day of P_1 can be used, and no more than \$7000 lb/day of P_2 can be used.

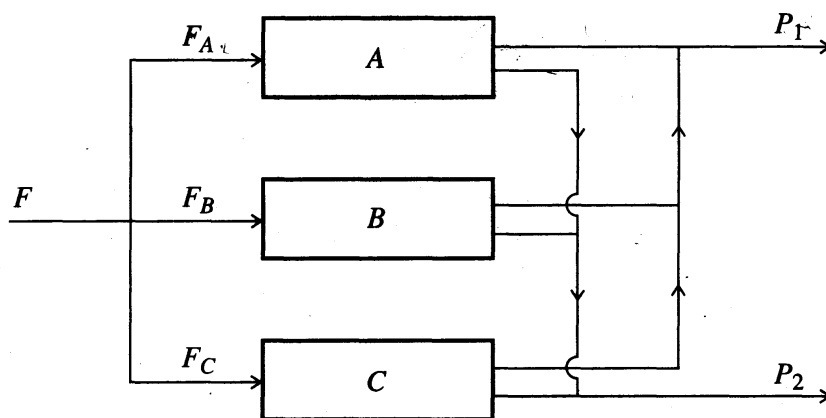


FIGURE P7.3

In order to determine the values of F_A , F_B , and F_C that maximize the daily profit, prepare a mathematical statement of this problem as a linear programming problem. Do *not* solve it.

- 7.4 Prepare a graph of the constraints and objective function, and solve the following linear programming problem

$$\begin{aligned}
 &\text{Maximize:} && x_1 + 2x_2 \\
 &\text{Subject to:} && -x_1 + 3x_2 < 10 \\
 &&& x_1 + x_2 \leq 6 \\
 &&& x_1 - x_2 \leq 2 \\
 &&& x_1 + 3x_2 \geq 6 \\
 &&& 2x_1 + x_2 \geq 4 \\
 &&& x_1 \geq 0 \quad x_2 \geq 0
 \end{aligned}$$

- 7.5 A chemical manufacturing firm has discontinued production of a certain unprofitable product line. This has created considerable excess production capacity on the three existing batch production facilities. Management is considering devoting this excess capacity to one or more of three new products: Call them products 1, 2, and 3. The available capacity on the existing units that might limit output is summarized in the following table:

Unit	Available time (h/week)
A	20
B	10
C	5

Each of the three new products requires the following processing time for completion:

Unit	Productivity (h/batch)		
	Product 1	Product 2	Product 3
A	0.8	0.2	0.3
B	0.4	0.3	
C	0.2		0.1

The sales department indicates that the sales potential for products 1 and 2 exceeds the maximum production rate and that the sales potential for product 3 is 20 batches per week. The profit per batch is \$20, \$6, and \$8, respectively, on products 1, 2, and 3.

Formulate a linear programming model for determining how much of each product the firm should produce to maximize profit.

- 7.6 An oil refinery has to blend gasoline. Suppose that the refinery wishes to blend four petroleum constituents into three grades of gasoline: A, B, and C. Determine the mix of the four constituents that will maximize profit.

The availability and costs of the four constituents are given in the following table:

Constituent*	Maximum quantity available (bbl/day)	Cost per barrel (\$)
X 1	3000	13.00
X 2	2000	15.30
X 3	4000	14.60
X 4	1000	14.90

- *1 = butane
 2 = straight-run
 3 = thermally cracked
 4 = catalytic cracked

To maintain the required quality for each grade of gasoline, it is necessary to specify certain maximum or minimum percentages of the constituents in each blend. These are shown in the following table, along with the selling price for each grade.

Grade	Specification	Selling price per barrel (\$)
A	Not more than 15% of 1 Not less than 40% of 2 Not more than 50% of 3	16.20
B	Not more than 10% of 1 Not less than 10% of 2	15.75
C	Not more than 20% of 1	15.30

Assume that all other cash flows are fixed so that the “profit” to be maximized is total sales income minus the total cost of the constituents. Set up a linear programming model for determining the amount and blend of each grade of gasoline.

- 7.7 A refinery produces, on average, 1000 gallon/hour of virgin pitch in its crude distillation operation. This pitch may be blended with flux stock to make commercial fuel oil, or it can be sent in whole or in part to a visbreaker unit as shown in Figure P7.7. The visbreaker produces an 80 percent yield of tar that can also be blended with flux stock to make commercial fuel oil. The visbreaking operation is economically break-even if the pitch and the tar are given no value, that is, the value of the overhead product equals the cost of the operation. The commercial fuel oil brings a realization of 5¢/gal, but the flux stock has a cracking value of 8¢/gal. This information together with the viscosity and gravity blending numbers and product specifications, appears in the following table. It is desired to operate for maximum profit.

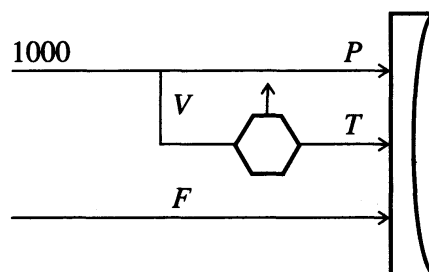


FIGURE P7.7

Fuel oil blending problem

	Quantity available (gal/h)	Value (¢/gal)	Viscosity Bl. No.	Gravity Bl. No.
Pitch	$P = 1000 - V$	0	5	8
Visbreaker feed	V	0	—	—
Tar	$T = 0.8V$	0	11	7
Flux	$F = \text{any}$	8	37	24
Fuel oil	$P + T + F$	5	21 min	12 min

Abbreviation: Bl. No. = blending number.

Formulate the preceding problem as a linear programming problem. How many variables are there? How many inequality constraints? How many equality constraints? How many bounds on the variables?

7.8 Examine the following problem:

$$\begin{aligned} \text{Minimize: } f &= 3x_1 + x_2 + x_3 \\ \text{Subject to: } x_1 - 2x_2 + x_3 &\leq 11 \\ -4x_1 + x_2 + 2x_3 &\geq 3 \\ 2x_1 &\quad - x_3 = -1 \\ x_1, x_2, x_3 &\geq 0 \end{aligned}$$

Is there a basic feasible solution to the problem? Answer yes or no, and explain.

7.9 An LP problem has been converted to standard canonical form by the addition of slack variables and has a basic feasible solution (with $x_1 = x_2 = 0$) as shown in the following set of equations:

$$\begin{aligned} -2x_1 + 2x_2 + x_3 &= 3 \\ 5x_1 + 2x_2 + x_4 &= 11 \\ x_1 + x_2 + x_5 &= 4 \\ 4x_1 + 2x_2 + f &= 0 \end{aligned}$$

Answer the following questions:

- Which variable should be increased first?
- Which row and which column designate the pivot point?
- What is the limiting value of the variable you designated part in (a)?

7.10 For the problem given in 7.9, find the next basis. Show the steps you take to calculate the improved solution, and indicate what the basic variables and nonbasic variables are in the new set of equations. (Just a single step from one vertex to the next is asked for in this problem.)

7.11 Examine the following problem

$$\begin{aligned} \text{Minimize: } f &= 3x_1 + x_2 + x_3 \\ \text{Subject to: } x_1 - 2x_2 + x_3 &\leq 11 \\ -4x_1 + x_2 + 2x_3 &\geq 3 \\ 2x_1 &\quad - x_3 = -1 \\ x_1, x_2, x_3 &\geq 0 \end{aligned}$$

Is there a basic feasible solution to the problem? Answer yes or no, and explain.

7.12 You are asked to solve the following problem:

$$\begin{aligned} \text{Maximize: } f &= 5x_1 + 2x_2 + 3x_3 \\ \text{Subject to: } x_1 + 2x_2 + 2x_3 + x_4 &= 8 \\ 3x_1 + 4x_2 + x_3 - x_5 &= 7 \\ x_1, \dots, x_5 &\geq 0 \end{aligned}$$

Explain in detail what you would do to obtain the first feasible solution to this problem. Show all equations. You do not have to calculate the feasible solution—just explain in detail how you would calculate it.

7.13 You are given the following LP equation sets:

$$\begin{aligned} \text{(a)} \quad & 3x_1 - x_2 + x_3 & & = -6 \\ & 4x_1 - 3x_2 & + x_4 & = -4 \\ & x_1 + 3x_2 & & + f = 0 \end{aligned}$$

Why is this formulation problematic?

$$\begin{aligned} \text{(b)} \quad & x_1 - 2x_2 + x_3 & & = 7 \\ & x_1 - 3x_2 & + x_4 & = 4 \\ & x_1 + 3x_2 & & + f = 0 \end{aligned}$$

Is the problem that leads to the preceding formulation solvable? How do you interpret this problem geometrically?

$$\begin{aligned} \text{(c)} \quad & 4x_1 + 2x_2 + x_3 & & = 6 \\ & 6x_1 + 3x_2 & + x_4 & = 9 \\ & x_1 + 3x_2 & & + f = 0 \end{aligned}$$

Apply the simplex rules to minimize f for the formulation. Is the solution unique?

$$\begin{aligned} \text{(d)} \quad & 4x_1 + 2x_2 + x_3 & & = 7 \\ & 6x_1 + 3x_2 & + x_4 & = 5 \\ & -x_1 & & + f = 0 \end{aligned}$$

Can you find the minimum of f ? Why or why not?

7.14 Solve the following LP:

$$\begin{aligned} \text{Minimize: } & f = x_1 + x_2 \\ \text{Subject to: } & x_1 + 3x_2 \leq 12 \\ & x_1 - x_2 \leq 1 \\ & 2x_1 - x_2 \leq 4 \\ & 2x_1 + x_2 \leq 8 \\ & x_1 \geq 0 \quad x_2 \geq 0 \end{aligned}$$

Does the solution via the simplex method exhibit cycling?

7.15 In Problem 7.1 what are the shadow prices for incremental production of gasoline, kerosene, and fuel oil? Suppose the profit coefficient for crude #1 is increased by 10 percent and crude #2 by 5 percent. Which change has a larger influence on the objective function?

7.16 For Problem 7.9, find the next basis. Show the steps for calculating the new table, and indicate the basic and nonbasic variables in the new table. (Just a single step from one vertex to the next is asked for in this problem.)

7.17 Solve the following linear programming problem:

$$\text{Maximize: } f = x_1 + 3x_2 - x_3$$

$$\text{Subject to: } x_1 + 2x_2 + x_3 = 4$$

$$2x_1 + x_2 \leq 5$$

7.18 Solve the following problem:

$$\text{Maximize: } f = 7x_1 + 12x_2 + 3x_3$$

$$\text{Subject to: } 2x_1 + 2x_2 + x_3 \leq 16$$

$$4x_1 + 8x_2 + x_3 \leq 40$$

$$x_1, x_2, x_3 \geq 0$$

7.19 Solve the following problem:

$$\text{Maximize: } f = 6x_1 + 5x_2$$

$$\text{Subject to: } 2x_1 + 5x_2 \leq 20$$

$$-5x_1 - x_2 \leq -5$$

$$-3x_1 - 11x_2 \leq -33$$

The following figure shows the constraints. If slack variables x_3 , x_4 and x_5 are added respectively to the inequality constraints, you can see from the diagram that the origin is not a feasible point, that is, you cannot start the simplex method by letting $x_1 = x_2 = 0$ because then $x_3 = 20$, $x_4 = -5$, and $x_5 = -33$, a violation of the assumption in linear programming that $x_i \geq 0$. What should you do to apply the simplex method to the problem other than start a phase I procedure of introducing artificial variables?

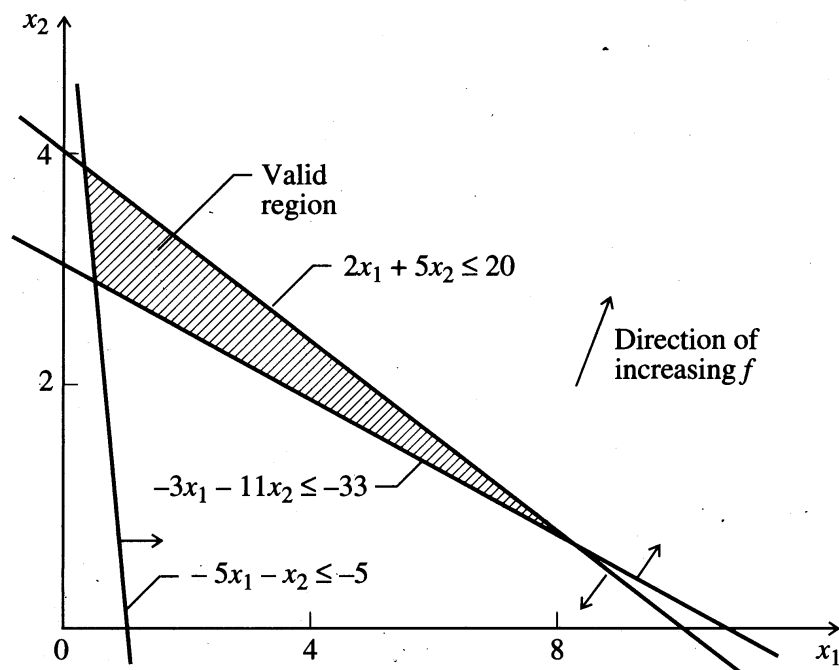


FIGURE P7.19

7.20 Are the following questions true or false and explain why:

- In applying the simplex method of linear programming, the solution found, if one is found, is the global solution to the problem.
- The solution to a linear programming problem is a unique solution.
- The solution to a linear programming problem that includes only inequality constraints (no equality constraints) never occurs in the interior of the feasible region.

7.21 A company has two alkylate plants, A_1 and A_2 , from which a given product is distributed to customers C_1 , C_2 , and C_3 . The transportation costs are given as follows:

Refinery	A_1	A_1	A_1	A_2	A_2	A_2
Customer	C_1	C_2	C_3	C_1	C_2	C_3
Cost (\$/ton)	25	60	75	20	50	85

The maximum refinery production rates and minimum customer demand rates are fixed and known to be as follows:

Customer or refinery	A_1	A_2	C_1	C_2	C_3
Rate (tons/day)	1.6	0.8	0.9	0.7	0.3

The cost of production for A_1 is \$30/ton for production levels less than 0.5 ton/day; for production levels greater than 0.5 ton/day, the production cost is \$40/ton. A_2 's production cost is uniform at \$35/ton.

Find the optimum distribution policy to minimize the company's total costs.

7.22 Alkylate, cat cracked gasoline, and straight run gasoline are blended to make aviation gasolines A and B and two grades of motor gasoline. The specifications on motor gasoline are not as rigid as for aviation gas. Physical property and production data for the inlet streams are as follows:

Stream	RVP	ON(0)	ON(4)	Available (bb/day)
Alkylate	5	94	108	4000
Cat cracked gasoline	8	84	94	2500
Straight run gasoline	4	74	86	4000

Abbreviations:

RVP = Reid vapor pressure (measure of volatility);

ON = octane number; in parentheses, number of mL/gal of tetraethyl lead (TEL).

For the blended products:

Product	RVP	TEL level	ON	Profit (\$/bbl)
Aviation gasoline A	≤ 7	0	≥ 80	5.00
Aviation gasoline B	≤ 7	4	≥ 91	5.50
Leaded motor gasoline	—	4	≥ 87	4.50
Unleaded motor gasoline	—	0	≥ 91	4.50

Set up this problem as an LP problem, and solve using a standard LP computer code.

7.23 A chemical plant makes three products and uses three raw materials in limited supply as shown in Figure P7.23. Each of the three products is produced in a separate process (1, 2, 3) according to the schematic shown in the figure.

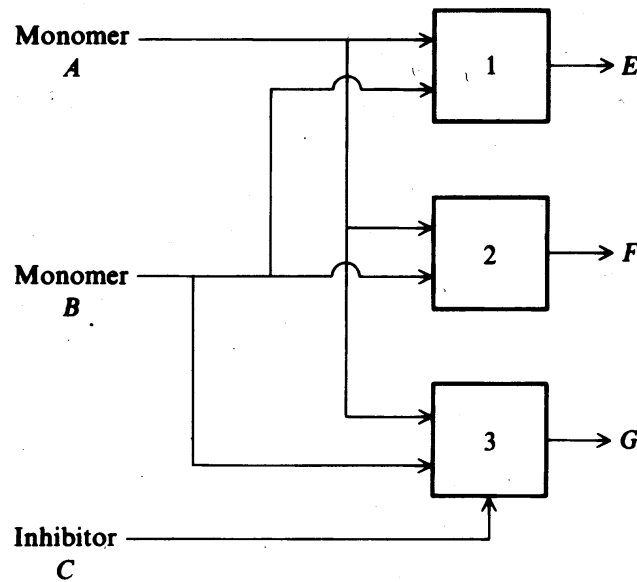


FIGURE P7.23

The available A , B , and C do not have to be totally consumed.

Process data:

Raw material	Maximum available (lb/day)	Cost (\$/100 lb)
A	4000	1.50
B	3000	2.00
C	2500	2.50

Process	Product	Reactants needed (lb/lb product)	Operating cost (\$)	Selling price of product (\$)
1	E	$\frac{2}{3}A, \frac{1}{3}B$	1.00/100 lb A (consumed in 1)	4.00/100 lb E
2	F	$\frac{2}{3}A, \frac{1}{3}B$	0.50/100 lb A (consumed in 2)	3.30/100 lb F
3	G	$\frac{1}{2}A, \frac{1}{6}B, \frac{1}{3}C$	1.00/100 lb G (produced in 3)	3.80/100 lb G

Set up the linear profit function and linear constraints to find the optimum product distribution, and apply the simplex technique to obtain numerical answers.

7.24 Ten grades of crude are available in the quantities shown in the table ranging from 10,000 to 30,000 barrels per day each, with an aggregate availability of 200,000 barrels per day. Refineries X , Y , and Z have incremental operations with stated requirements totaling 180,000 barrels per day. Of the available crude, 20,000 barrels per day is not used. One of the refineries can operate at two incremental operations, X_1 and X_2 , which represent different efficiency levels. The net profit or loss for each crude in each refinery operation

is given in the table in cents per barrel. It is assumed that the crude evaluations reflect the resulting product distribution from these incremental operations. (In practice, however, if further debits are encountered in the solution because of lack of product quality or for transportation of surplus products, suitable corrections can be made in the crude evaluations and the problem reworked until a realistic solution is obtained.)

Maximize the profit per day by allocating the ten crudes among the three refineries with X being able to operate at two levels, so specify X_1 and X_2 as well as Y and Z .

Crude evaluation, availability and requirement

Crude	a	b	c	d	e	f	g	h	i	j	Required (M bpd)
	(Profit or loss of each refinery cpb)										
Refinery											
X_1	-6	3	17	10	63	34	15	22	-2	15	30
X_2	-11	-7	-16	9	49	16	4	10	-8	8	40
Y	-7	3	16	13	60	25	12	19	4	13	50
Z	-1	0	13	3	48	15	7	17	9	3	60
Available (M bpd)	30	30	20	20	10	20	20	10	30	10	200

Abbreviations: $M = 1000$; bpd = barrels per day; cpb = cents per barrel.

7.25 Consider a typical linear programming example in which N grades of paper are produced on a paper machine. Due to raw materials restrictions not more than a_i tons of grade i can be produced in a week. Let

- x_i = numbers of tons of grade i produced during the week
- b_i = number of hours required to produce a ton of grade i
- p_i = profit made per ton of grade i

Because 160 production hours are available each week, the problem is to find non-negative values of $x_i, i = 1, \dots, N$, and the integer value N that satisfy

$$x_i \leq a_i \tag{1}$$

$$\sum_{i=1}^N b_i x_i \leq c \tag{2}$$

and that maximize the profit function

$$f(x_1, \dots, x_N) = \sum_{i=1}^N p_i x_i \tag{3}$$

Data:

	a_i	b_i	p_i	$c = 160$
1	400	0.2	20	
2	300	0.4	50	
3	200	0.2	20	
4	100	0.2	10	
5	50	0.2	10	

NONLINEAR PROGRAMMING WITH CONSTRAINTS

8.1	Direct Substitution	265
8.2	First-Order Necessary Conditions for a Local Extremum	267
8.3	Quadratic Programming	284
8.4	Penalty, Barrier, and Augmented Lagrangian Methods	285
8.5	Successive Linear Programming	293
8.6	Successive Quadratic Programming	302
8.7	The Generalized Reduced Gradient Method	306
8.8	Relative Advantages and Disadvantages of NLP Methods	318
8.9	Available NLP Software	319
8.10	Using NLP Software	323
	References	328
	Supplementary References	329
	Problems	329

CHAPTER 1 PRESENTS some examples of the constraints that occur in optimization problems. Constraints are classified as being inequality constraints or equality constraints, and as linear or nonlinear. Chapter 7 described the simplex method for solving problems with linear objective functions subject to linear constraints. This chapter treats more difficult problems involving minimization (or maximization) of a nonlinear objective function subject to linear or nonlinear constraints:

$$\begin{aligned} \text{Minimize: } & f(\mathbf{x}) & \mathbf{x} &= [x_1 \ x_2 \ \cdots \ x_n]^T \\ \text{Subject to: } & h_i(\mathbf{x}) = b_i & i &= 1, 2, \dots, m \\ & g_j(\mathbf{x}) \leq c_j & j &= 1, \dots, r \end{aligned} \quad (8.1)$$

The inequality constraints in Problem (8.1) can be transformed into equality constraints as explained in Section 8.4, so we focus first on problems involving only equality constraints.

8.1 DIRECT SUBSTITUTION

One method of handling just one or two linear or nonlinear equality constraints is to solve explicitly for one variable and eliminate that variable from the problem formulation. This is done by direct substitution in the objective function and constraint equations in the problem. In many problems elimination of a single equality constraint is often superior to an approach in which the constraint is retained and some constrained optimization procedure is executed. For example, suppose you want to minimize the following objective function that is subject to a single equality constraint

$$\text{Minimize: } f(\mathbf{x}) = 4x_1^2 + 5x_2^2 \quad (8.2a)$$

$$\text{Subject to: } 2x_1 + 3x_2 = 6 \quad (8.2b)$$

Either x_1 or x_2 can be eliminated without difficulty. Solving for x_1 ,

$$x_1 = \frac{6 - 3x_2}{2} \quad (8.3)$$

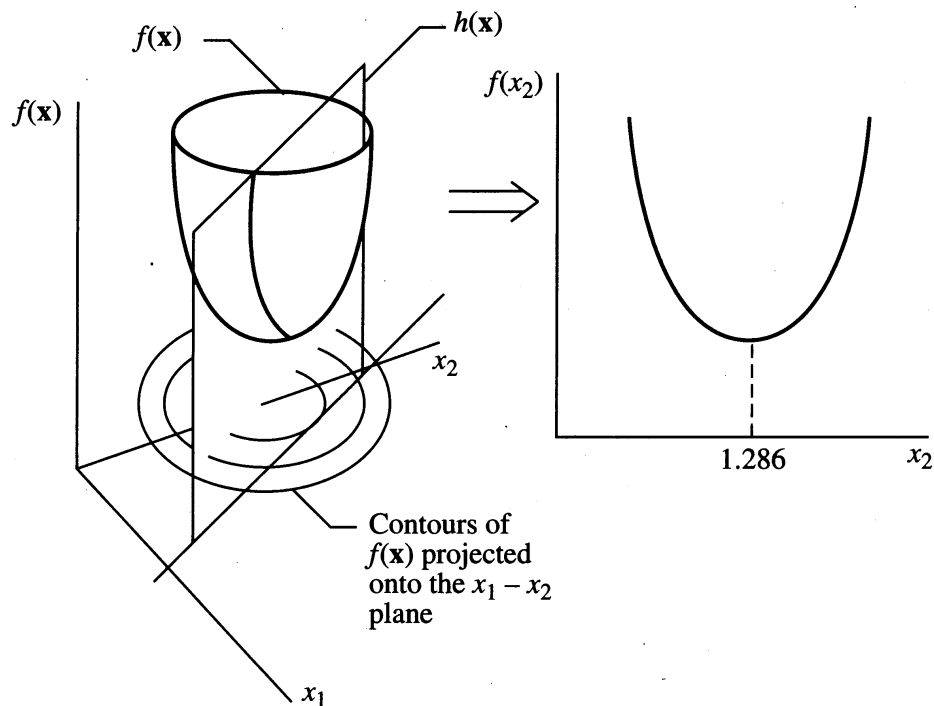
we can substitute for x_1 in Equation (8.2a). The new equivalent objective function in terms of a single variable x_2 is

$$f(x_2) = 14x_2^2 - 36x_2 + 36 \quad (8.4)$$

The constraint in the original problem has now been eliminated, and $f(x_2)$ is an unconstrained function with 1 degree of freedom (one independent variable). Using constraints to eliminate variables is the main idea of the generalized reduced gradient method, as discussed in Section 8.7.

We can now minimize the objective function (8.4), by setting the first derivative of f equal to zero, and solving for the optimal value of x_2 :

$$\frac{df(x_2)}{dx_2} = 28x_2 - 36 = 0 \quad x_2^* = 1.286$$

**FIGURE 8.1**

Graphical representation of a function of two variables reduced to a function of one variable by direct substitution. The unconstrained minimum is at $(0,0)$, the center of the contours.

Once x_2^* is obtained, then, x_1^* can be directly obtained via the constraint (8.2b):

$$x_1^* = \frac{6 - 3x_2^*}{2} = 1.071$$

The geometric interpretation for the preceding problem requires visualizing the objective function as the surface of a paraboloid in three-dimensional space, as shown in Figure 8.1. The projection of the intersection of the paraboloid and the plane representing the constraint onto the $f(x_2) = x_2$ plane is a parabola. We then find the minimum of the resulting parabola. The elimination procedure described earlier is tantamount to projecting the intersection locus onto the x_2 axis. The intersection locus could also be projected onto the x_1 axis (by elimination of x_2). Would you obtain the same result for \mathbf{x}^* as before?

In problems in which there are n variables and m equality constraints, we could attempt to eliminate m variables by direct substitution. If all equality constraints can be removed, and there are no inequality constraints, the objective function can then be differentiated with respect to each of the remaining $(n - m)$ variables and the derivatives set equal to zero. Alternatively, a computer code for unconstrained optimization can be employed to obtain \mathbf{x}^* . If the objective function is convex (as in the preceding example) and the constraints form a convex region, then any stationary point is a global minimum. Unfortunately, very few problems in practice assume this simple form or even permit the elimination of all equality constraints.

Consequently, in this chapter we will discuss five major approaches for solving nonlinear programming problems with constraints:

1. Analytic solution by solving the first-order necessary conditions for optimality (Section 8.2)
2. Penalty and barrier methods (Section 8.4)
3. Successive linear programming (Section 8.5)
4. Successive quadratic programming (Section 8.6)
5. Generalized reduced gradient (Section 8.7)

The first of these methods is usually only suitable for small problems with a few variables, but it can generate much useful information and insight when it is applicable. The others are numerical approaches, which must be implemented on a computer.

8.2 FIRST-ORDER NECESSARY CONDITIONS FOR A LOCAL EXTREMUM

As an introduction to this subject, consider the following example.

EXAMPLE 8.1 GRAPHIC INTERPRETATION OF A CONSTRAINED OPTIMIZATION PROBLEM

$$\text{Minimize: } f(x_1, x_2) = x_1 + x_2$$

$$\text{Subject to: } h(x_1, x_2) = x_1^2 + x_2^2 - 1 = 0$$

Solution. This problem is illustrated graphically in Figure E8.1a. Its feasible region is a circle of radius one. Contours of the linear objective $x_1 + x_2$ are lines parallel to the one in the figure. The contour of lowest value that contacts the circle touches it at the point $\mathbf{x}^* = (-0.707, -0.707)$, which is the global minimum. You can solve this problem analytically as an unconstrained problem by substituting for x_1 or x_2 by using the constraint.

Certain relations involving the gradients of f and h hold at \mathbf{x}^* if \mathbf{x}^* is a local minimum. These gradients are

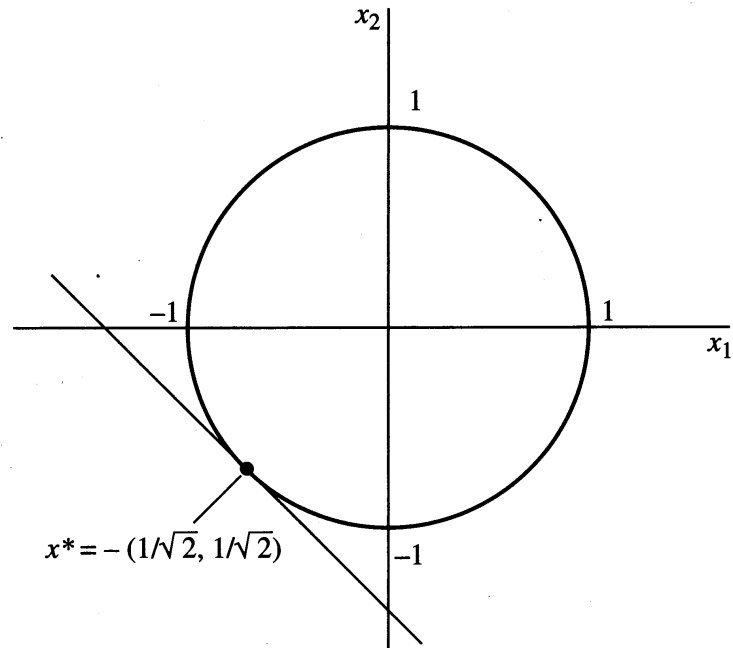
$$\nabla f(\mathbf{x}^*) = [1, 1]$$

$$\nabla h(\mathbf{x}^*) = [2x_1, 2x_2]|_{\mathbf{x}^*} = [-1.414, -1.414]$$

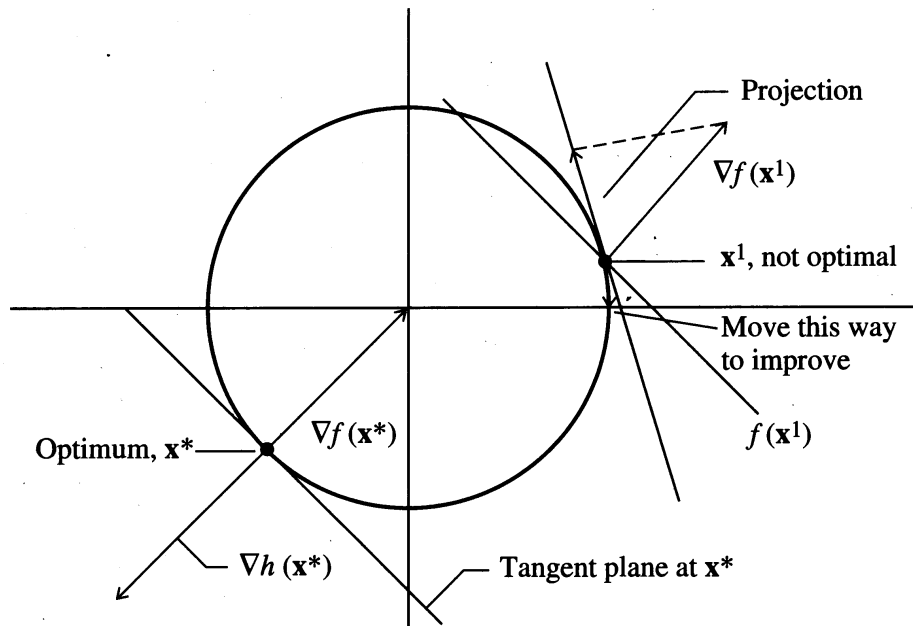
and are shown in Figure E8.1b. The gradient of the objective function $\nabla f(\mathbf{x}^*)$ is orthogonal to the tangent plane of the constraint at \mathbf{x}^* . In general $\nabla h(\mathbf{x}^*)$ is *always* orthogonal to this tangent plane, hence $\nabla f(\mathbf{x}^*)$ and $\nabla h(\mathbf{x}^*)$ are collinear, that is, they lie on the same line but point in opposite directions. This means the two vectors must be multiples of each other;

$$\nabla f(\mathbf{x}^*) = \lambda^* \nabla h(\mathbf{x}^*) \tag{a}$$

where $\lambda^* = -1/1.414$ is called the *Lagrange multiplier* for the constraint $h = 0$.

**FIGURE E8.1a**

Circular feasible region with objective function contours and the constraint.

**FIGURE E8.1b**

Gradients at the optimal point and at a nonoptimal point.

The relationship in Equation (a) *must* hold at *any* local optimum of *any* equality-constrained NLP involving smooth functions. To see why, consider the nonoptimal point \mathbf{x}^1 in Figure E8.1b. $\nabla f(\mathbf{x}^1)$ is *not* orthogonal to the tangent plane of the constraint at \mathbf{x}^1 , so it has a nonzero projection on the plane. The negative of this projected gradient is also nonzero, indicating that moving downward along the circle reduces

(improves) the objective function. At a local optimum, no small or incremental movement along the constraint (the circle in this problem) away from the optimum can improve the value of the objective function, so the projected gradient must be zero. This can only happen when $\nabla f(\mathbf{x}^*)$ is orthogonal to the tangent plane.

The relation (a) in Example 8.1 can be rewritten as

$$\nabla f(\mathbf{x}^*) + \lambda^* \nabla h(\mathbf{x}^*) = 0 \quad (8.5)$$

where $\lambda^* = 0.707$. We now introduce a new function $L(\mathbf{x}, \lambda)$ called the Lagrangian function:

$$L(\mathbf{x}, \lambda) = f(\mathbf{x}) + \lambda h(\mathbf{x}) \quad (8.6)$$

Then Equation (8.5) becomes

$$\nabla_{\mathbf{x}} L(\mathbf{x}, \lambda)|_{(\mathbf{x}^*, \lambda^*)} = 0 \quad (8.7)$$

so the gradient of the Lagrangian function with respect to \mathbf{x} , evaluated at $(\mathbf{x}^*, \lambda^*)$, is zero. Equation (8.7), plus the feasibility condition

$$h(\mathbf{x}^*) = 0 \quad (8.8)$$

constitute the first-order necessary conditions for optimality. The scalar λ is called a *Lagrange multiplier*.

Using the necessary conditions to find the optimum

The first-order necessary conditions (8.7) and (8.8) can be used to find an optimal solution. Assume \mathbf{x}^* and λ^* are unknown. The Lagrangian function for the problem in Example 8.1 is

$$L(\mathbf{x}, \lambda) = x_1 + x_2 + \lambda(x_1^2 + x_2^2 - 1)$$

Setting the first partial derivatives of L with respect to \mathbf{x} to zero, we get

$$\frac{\partial L}{\partial x_1} = 1 + 2\lambda x_1 = 0 \quad (8.9)$$

$$\frac{\partial L}{\partial x_2} = 1 + 2\lambda x_2 = 0 \quad (8.10)$$

The feasibility condition (8.8) is

$$x_1^2 + x_2^2 - 1 = 0 \quad (8.11)$$

The first-order necessary conditions for this problem, Equations (8.9)–(8.11), consist of three equations in three unknowns (x_1, x_2, λ) . Solving (8.9)–(8.10) for x_1 and x_2 gives

$$x_1 = x_2 = -\frac{1}{2\lambda} \quad (8.12)$$

which shows that x_1 and x_2 are equal at the extremum. Substituting Equation (8.12) into Equation (8.11);

$$\frac{1}{4\lambda^2} + \frac{1}{4\lambda^2} = 1$$

or

$$2\lambda^2 = 1 \quad (8.13)$$

so

$$\lambda = \pm 0.707$$

and

$$x_1 = x_2 = \mp 0.707 \quad (8.14)$$

The minus sign corresponds to the minimum of f , and the plus sign to the maximum.

EXAMPLE 8.2 USE OF LAGRANGE MULTIPLIERS

Consider the problem introduced earlier in Equation (8.2):

$$\text{Minimize: } f(\mathbf{x}) = 4x_1^2 + 5x_2^2 \quad (a)$$

$$\text{Subject to: } h(\mathbf{x}) = 0 = 2x_1 + 3x_2 - 6 \quad (b)$$

Solution. Let

$$L(\mathbf{x}, \lambda) = 4x_1^2 + 5x_2^2 + \lambda(2x_1 + 3x_2 - 6) \quad (c)$$

Apply the necessary conditions (8.11) and (8.12)

$$\frac{\partial L(\mathbf{x}, \lambda)}{\partial x_1} = 8x_1 + 2\lambda = 0 \quad (d)$$

$$\frac{\partial L(\mathbf{x}, \lambda)}{\partial x_2} = 10x_2 + 3\lambda = 0 \quad (e)$$

$$\frac{\partial L(\mathbf{x}, \lambda)}{\partial \lambda} = 2x_1 + 3x_2 - 6 = 0 \quad (f)$$

By substitution, $x_1 = -\lambda/4$ and $x_2 = -3\lambda/10$, and therefore Equation (f) becomes

$$2\left(\frac{-\lambda}{4}\right) + 3\left(\frac{-3\lambda}{10}\right) - 6 = 0$$

$$\lambda^* = -4.286$$

$$x_1^* = 1.071$$

$$x_2^* = 1.286$$

8.2.1 Problems Containing Only Equality Constraints

A general equality constrained NLP with m constraints and n variables can be written as

$$\text{Maximize: } f(\mathbf{x}) \quad (8.15)$$

$$\text{Subject to: } h_j(\mathbf{x}) = b_j, \quad j = 1, \dots, m$$

where $\mathbf{x} = (x_1, \dots, x_n)$ is the vector of decision variables, and each b_j is a constant. We assume that the objective f and constraint functions h_j have continuous first partial derivatives. Corresponding to each constraint $h_j = b_j$, define a Lagrange multiplier λ_j and let $\boldsymbol{\lambda} = (\lambda_1, \dots, \lambda_m)$ be the vector of these multipliers. The Lagrangian function for the problem is

$$L(\mathbf{x}, \boldsymbol{\lambda}) = f(\mathbf{x}) + \sum_{j=1}^m \lambda_j [h_j(\mathbf{x}) - b_j] \quad (8.16)$$

and the first-order necessary conditions are

$$\frac{\partial L}{\partial x_i} = \frac{\partial f}{\partial x_i} + \sum_{j=1}^m \lambda_j \frac{\partial h_j}{\partial x_i} = 0, \quad i = 1, \dots, n \quad (8.17)$$

$$h_j(\mathbf{x}) = b_j, \quad j = 1, \dots, m \quad (8.18)$$

Note that there are $n + m$ equations in the $n + m$ unknowns \mathbf{x} and $\boldsymbol{\lambda}$. In Section 8.6 we describe an important class of NLP algorithms called successive quadratic programming (SQP), which solve (8.17)–(8.18) by a variant of Newton's method.

Problem (8.15) must satisfy certain conditions, called constraint qualifications, in order for Equations (8.17)–(8.18) to be applicable. One constraint qualification (see Luenberger, 1984) is that the gradients of the equality constraints, evaluated at \mathbf{x}^* , should be linearly independent. Now we can state formally the first order necessary conditions.

First-order necessary conditions for an extremum

Let \mathbf{x}^ be a local minimum or maximum for the problem (8.15), and assume that the constraint gradients $\nabla h_j(\mathbf{x}^*)$, $j = 1, \dots, m$, are linearly independent. Then there exists a vector of Lagrange multipliers $\boldsymbol{\lambda}^* = (\lambda_1^*, \dots, \lambda_m^*)$ such that $(\mathbf{x}^*, \boldsymbol{\lambda}^*)$ satisfies the first-order necessary conditions (8.17)–(8.18).*

Examples illustrating what can go wrong if the constraint gradients are dependent at \mathbf{x}^* can be found in Luenberger (1984). It is important to remember that *all* local maxima and minima of an NLP satisfy the first-order necessary conditions if the constraint gradients at each such optimum are independent. Also, because these conditions are necessary but not, in general, sufficient, a solution of Equations (8.17)–(8.18) need *not* be a minimum or a maximum at all. It can be a saddle or inflection point. This is exactly what happens in the unconstrained case, where there are no constraint functions $h_j = 0$. Then conditions (8.17)–(8.18) become

$$\nabla f(\mathbf{x}) = \mathbf{0}$$

the familiar condition that the gradient must be zero (see Section 4.5). To tell if a point satisfying the first-order necessary conditions is a minimum, maximum, or neither, second-order sufficiency conditions are needed. These are discussed later in this section.

Sensitivity interpretation of Lagrange multipliers

Sensitivity analysis in NLP indicates how an optimal solution changes as the problem data change. These data include *any* parameters that appear in the objective or constraint functions, or on the right-hand sides of constraints. The Lagrange multipliers λ^* provide useful information on right-hand side changes, just as they do for linear programs (which are a special class of NLPs). To illustrate their application in NLP, consider again Example 8.1, with the constraint right-hand side (the square of the radius of the circle) treated as a parameter b ;

$$\begin{aligned} \text{Minimize: } & x_1 + x_2 \\ \text{Subject to: } & x_1^2 + x_2^2 = b \end{aligned}$$

The optimal solution of this problem is a function of b , denoted by $(x_1(b), x_2(b))$, as is the optimal multiplier value, $\lambda(b)$. Using the first-order necessary conditions (8.9)–(8.11), rewritten here as

$$\begin{aligned} 1 + 2\lambda x_1 &= 0 \\ 1 + 2\lambda x_2 &= 0 \\ x_1^2 + x_2^2 &= b \end{aligned}$$

The solution of these equations is (check it!);

$$\begin{aligned} x_1^*(b) = x_2^*(b) &= -\left(\frac{b}{2}\right)^{1/2} \\ \lambda^*(b) &= (2b)^{-1/2} \end{aligned}$$

These formulas agree with the previous results for $b = 1$. The minimal objective value, sometimes called the *optimal value function*, is

$$V(b) = x_1(b) + x_2(b) = -(2b)^{1/2}$$

The derivative of the optimal value function is

$$\frac{dV}{db} = -(2b)^{-1/2} = -\lambda^*(b)$$

so the negative of the optimal Lagrange multiplier value is dV/db . Hence, if we solve this problem for a specific b (for example $b = 1$) then the optimal objective value for b close to 1 has the first-order Taylor series approximation

$$\begin{aligned} V(b) &\approx V(1) - \lambda(1)(b - 1) \\ &= -\sqrt{2} - \frac{1}{\sqrt{2}}(b - 1) \end{aligned}$$

To see how useful these Lagrange multipliers are, consider the general problem (8.15), with right-hand sides b_i ;

$$\begin{aligned} \text{Minimize: } & f(\mathbf{x}) \\ \text{Subject to: } & h_i(\mathbf{x}) = b_i, \quad i = 1, \dots, m \end{aligned} \quad (8.19)$$

Let $\mathbf{b} = (b_1, \dots, b_m)$ be the right-hand side (rhs) vector, and $V(\mathbf{b})$ the optimal objective value. If $\bar{\mathbf{b}}$ is a specific right-hand side vector, and $(\mathbf{x}(\bar{\mathbf{b}}), \boldsymbol{\lambda}(\bar{\mathbf{b}}))$ is a local optimum for $\mathbf{b} = \bar{\mathbf{b}}$, then

$$-\lambda_j(\bar{\mathbf{b}}) = \left. \frac{\partial V}{\partial b_j} \right|_{\bar{\mathbf{b}}} \quad (8.20)$$

The constraints with the largest absolute λ_j values are the ones whose right-hand sides affect the optimal value function V the most, at least for b close to $\bar{\mathbf{b}}$. However, one must account for the units for each b_j in interpreting these values. For example, if some b_j is measured in kilograms and both sides of the constraint $h_j(\mathbf{x}) = b_j$ are multiplied by 2.2, then the new constraint has units of pounds, and its new Lagrange multiplier is $1/2.2$ times the old one.

8.2.2 Problems Containing Only Inequality Constraints

The first-order necessary conditions for problems with inequality constraints are called the *Kuhn–Tucker conditions* (also called Karush–Kuhn–Tucker conditions). The idea of a cone aids the understanding of the Kuhn–Tucker conditions (KTC). A cone is a set of points R such that, if \mathbf{x} is in R , $\boldsymbol{\lambda}^T \mathbf{x}$ is also in R for $\boldsymbol{\lambda} \geq 0$. A *convex cone* is a cone that is a convex set. An example of a convex cone in two dimensions is shown in Figure 8.2. In two and three dimensions, the definition of a convex cone coincides with the usual meaning of the word.

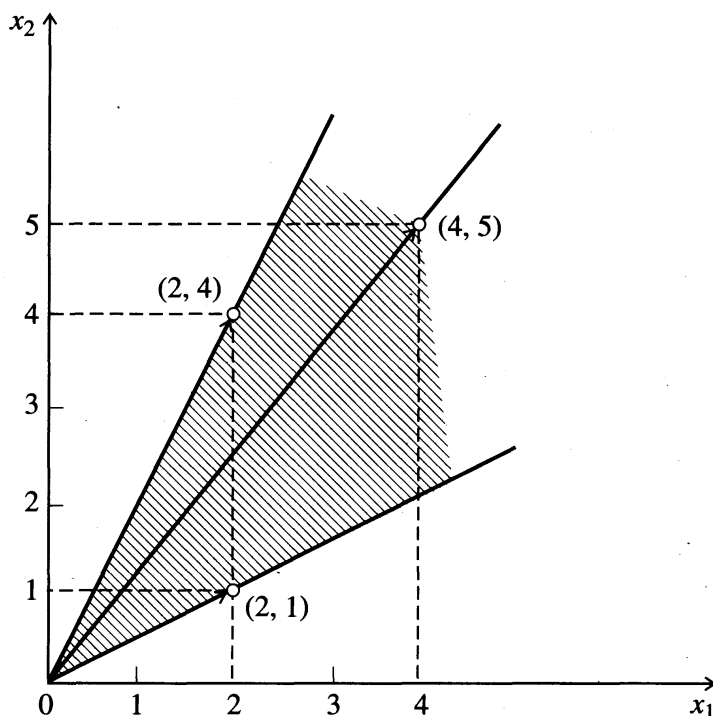
It can be shown from the preceding definitions that the set of all nonnegative linear combinations of a finite set of vectors is a convex cone, that is, that the set

$$R = \{\mathbf{x} \mid \mathbf{x} = \lambda_1 \mathbf{x}_1 + \lambda_2 \mathbf{x}_2 + \dots + \lambda_m \mathbf{x}_m, \lambda_i \geq 0, \quad i = 1, \dots, m\}$$

is a convex cone. The vectors $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_m$ are called the generators of the cone. For example, the cone of Figure 8.2 is generated by the vectors $[2, 1]$ and $[2, 4]$. Thus any vector that can be expressed as a nonnegative linear combination of these vectors lies in the cone. In Figure 8.2 the vector $[4, 5]$ in the cone is given by $[4, 5] = 1 \times [2, 1] + 1 \times [2, 4]$.

Kuhn–Tucker conditions: Geometrical interpretation

The Kuhn–Tucker conditions are predicated on this fact: At any local constrained optimum, no (small) allowable change in the problem variables can improve the value of the objective function. To illustrate this statement, consider the nonlinear programming problem:

**FIGURE 8.2**

The shaded region forms a convex cone.

$$\text{Minimize: } f(x,y) = (x - 2)^2 + (y - 1)^2$$

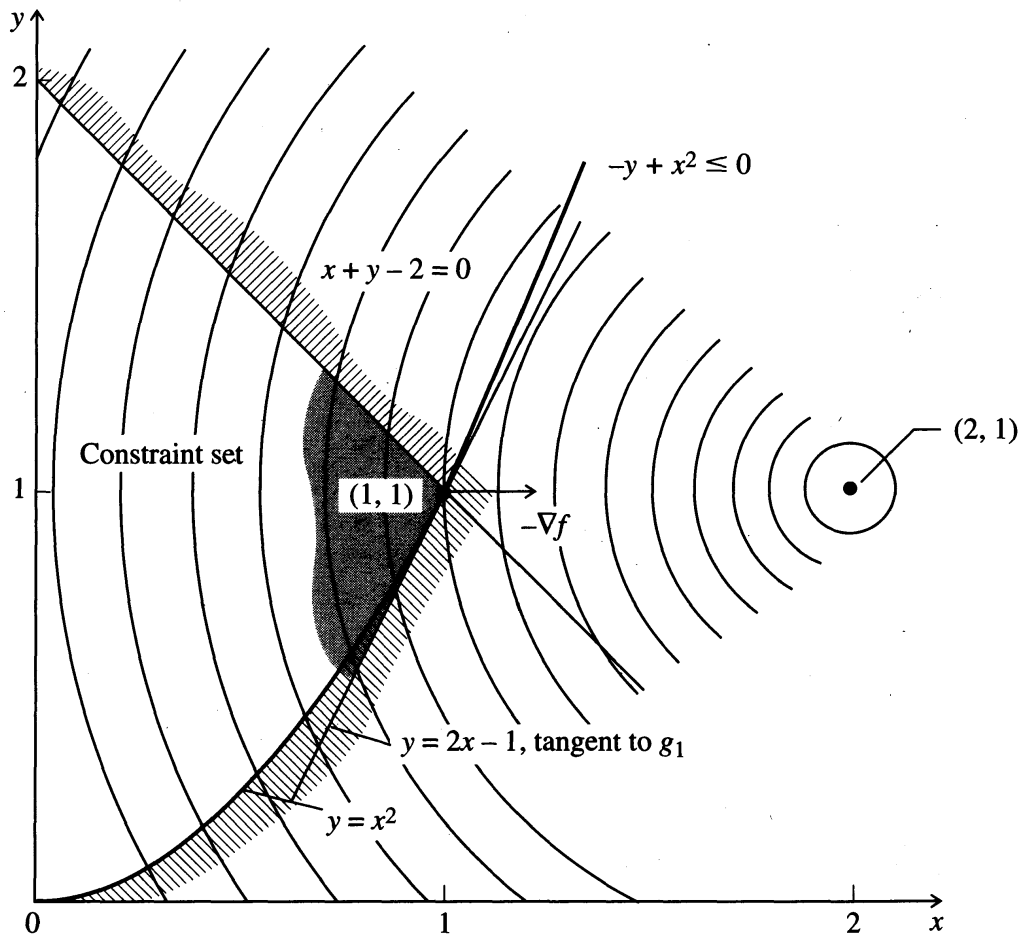
$$\text{Subject to: } g_1(x,y) = -y + x^2 \leq 0$$

$$g_2(x,y) = x + y \leq 2$$

$$g_3(x,y) = y \geq 0.$$

The problem is shown geometrically in Figure 8.3. It is evident that the optimum is at the intersection of the first two constraints at $(1, 1)$. Because these inequality constraints hold as equalities at $(1, 1)$, they are called *binding*, or *active*, constraints at this point. The third constraint holds as a strict inequality at $(1, 1)$, and it is an *inactive*, or *nonbinding*, constraint at this point. Define a *feasible direction* of search as a vector such that a differential move along that vector violates no constraints. At $(1, 1)$, the set of all feasible directions lies between the line $x + y - 2 = 0$ and the tangent line to $y = x^2$ at $(1, 1)$, that is, the line $y = 2x - 1$. In other words, the set of feasible directions is the cone generated by these lines that are shaded in the figure. The vector $-\nabla f$ points in the direction of the maximum rate of decrease of f , and a small move along any direction making an angle (defined as positive) of less than 90° with $-\nabla f$ will decrease f . Thus, at the optimum, no feasible direction can have an angle of less than 90° between it and $-\nabla f$.

Now consider Figure 8.4, in which the gradient vectors ∇g_1 and ∇g_2 are drawn. Note that $-\nabla f$ is contained in the cone generated by ∇g_1 and ∇g_2 . What if this were not so? If $-\nabla f$ were slightly above ∇g_2 , it would make an angle of less than 90° with a feasible direction just below the line $x + y - 2 = 0$. If $-\nabla f$ were slightly below ∇g_1 , it would make an angle of less than 90° with a feasible direction just

**FIGURE 8.3**

Geometry of a constrained optimization problem. The feasible region lies within the binding constraints plus the boundaries themselves.

above the line $y = 2x - 1$. Neither case can occur at an optimal point, and both cases are excluded if and only if $-\nabla f$ lies within the cone generated by ∇g_1 and ∇g_2 . Of course, this is the same as requiring that ∇f lie within the cone generated by $-\nabla g_1$ and $-\nabla g_2$. This leads to the usual statement of the KTC; that is, if f and all g_i are differentiable, a necessary condition for a point \mathbf{x}^* to be a constrained minimum of the problem

$$\text{Minimize: } f(\mathbf{x})$$

$$\text{Subject to: } g_j(\mathbf{x}) \leq c_j, \quad j = 1, \dots, r$$

is that, at \mathbf{x}^* , ∇f lies within the cone generated by the negative gradients of the binding constraints.

Algebraic statement of the Kuhn–Tucker conditions

The preceding results may be stated in algebraic terms. For ∇f to lie within the cone described earlier, it must be a nonnegative linear combination of the negative gradients of the binding constraints; that is, there must exist Lagrange multipliers u_j^* such that

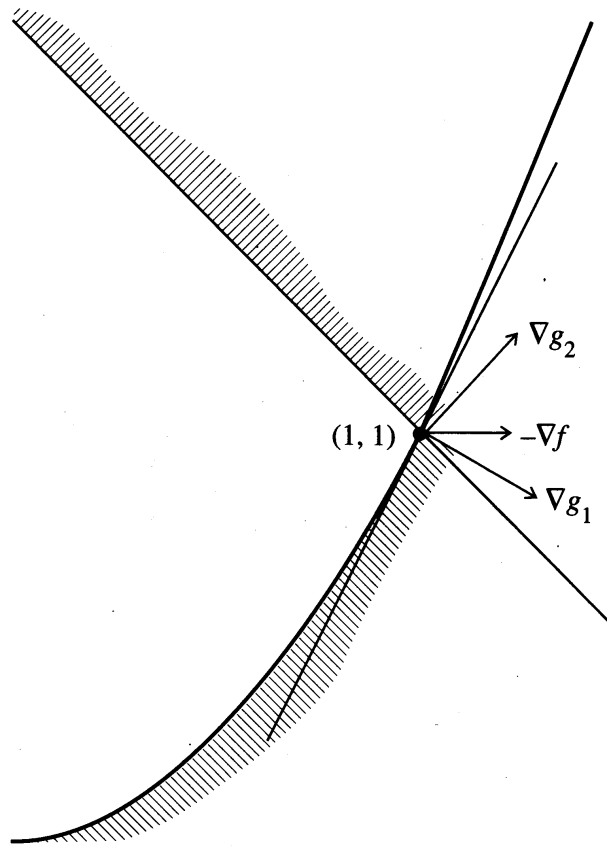


FIGURE 8.4
Gradient of objective contained in convex cone.

$$\nabla f(\mathbf{x}^*) = \sum_{j \in I} u_j^* [-\nabla g_j(\mathbf{x}^*)] \quad (8.21)$$

where

$$u_j^* \geq 0, \quad j \in I \quad (8.22)$$

and I is the set of indices of the binding inequality constraints. The multipliers u_j^* are analogous to λ_i defined for equality constraints.

These results may be restated to include all constraints by defining the multiplier u_j^* to be zero if $g_j(\mathbf{x}^*) < c_j$. In the previous example u_3^* , the multiplier of the inactive constraint g_3 , is zero. Then we can say that $u_j^* \geq 0$ if $g_j(\mathbf{x}^*) = c_j$, and $u_j^* = 0$ if $g_j(\mathbf{x}^*) < c_j$, thus the product $u_j^*[g_j(\mathbf{x}) - c_j]$ is zero for all j . This property, that inactive inequality constraints have zero multipliers, is called *complementary slackness*. Conditions (8.21) and (8.22) then become

$$\nabla f(\mathbf{x}^*) + \sum_{j=1}^r u_j^* \nabla g_j(\mathbf{x}^*) = 0 \quad (8.23)$$

$$u_j^* \geq 0, \quad u_j^*[g_j(\mathbf{x}^*) - c_j] = 0 \quad (8.24a)$$

$$g_j(\mathbf{x}^*) \leq c_j, \quad j = 1, \dots, r \quad (8.24b)$$

Relations (8.23) and (8.24) are the form in which the Kuhn–Tucker conditions are usually stated.

Lagrange multipliers

The KTC are closely related to the classical Lagrange multiplier results for equality constrained problems. Form the Lagrangian

$$L(\mathbf{x}, \mathbf{u}) = f(\mathbf{x}) + \sum_{j=1}^r u_j [g_j(\mathbf{x}) - c_j]$$

where the u_j are viewed as Lagrange multipliers for the inequality constraints $g_j(\mathbf{x}) \leq c_j$. Then Equations (8.23) and (8.24) state that $L(\mathbf{x}, \mathbf{u})$ must be stationary in \mathbf{x} at $(\mathbf{x}^*, \mathbf{u}^*)$ with the multipliers \mathbf{u}^* satisfying Equation (8.24). The stationarity of L is the same condition as in the equality-constrained case. The additional conditions in Equation (8.24) arise because the constraints here are inequalities.

8.2.3 Problems Containing both Equality and Inequality Constraints

When both equality and inequality constraints are present, the KTC are stated as follows: Let the problem be

$$\text{Minimize: } f(\mathbf{x}) \quad (8.25)$$

$$\text{Subject to: } h_i(\mathbf{x}) = b_i, \quad i = 1, \dots, m \quad (8.26a)$$

and

$$g_j(\mathbf{x}) \leq c_j, \quad j = 1, \dots, r \quad (8.26b)$$

Define Lagrange multipliers λ_i associated with the equalities and u_j for the inequalities, and form the Lagrangian function

$$L(\mathbf{x}, \boldsymbol{\lambda}, \mathbf{u}) = f(\mathbf{x}) + \sum_{i=1}^m \lambda_i [h_i(\mathbf{x}) - b_i] + \sum_{j=1}^r u_j [g_j(\mathbf{x}) - c_j] \quad (8.27)$$

Then, if \mathbf{x}^* is a local minimum of the problems (8.25)–(8.26), there exist vectors of Lagrange multipliers $\boldsymbol{\lambda}^*$ and \mathbf{u}^* , such that \mathbf{x}^* is a stationary point of the function $L(\mathbf{x}, \boldsymbol{\lambda}^*, \mathbf{u}^*)$, that is,

$$\nabla_{\mathbf{x}} L(\mathbf{x}^*, \boldsymbol{\lambda}^*, \mathbf{u}^*) = \nabla f(\mathbf{x}^*) + \sum_{i=1}^m \lambda_i^* \nabla h_i(\mathbf{x}^*) + \sum_{j=1}^r u_j^* \nabla g_j(\mathbf{x}^*) = \mathbf{0} \quad (8.28)$$

and complementary slackness hold for the inequalities:

$$u_j^* \geq 0 \quad u_j^* [g_j(\mathbf{x}^*) - c_j] = 0, \quad j = 1, \dots, r \quad (8.29)$$

EXAMPLE 8.3 APPLICATION OF THE LAGRANGE MULTIPLIER METHOD WITH NONLINEAR INEQUALITY CONSTRAINTS

Solve the problem

$$\text{Minimize: } f(\mathbf{x}) = x_1 x_2$$

$$\text{Subject to: } g(\mathbf{x}) = x_1^2 + x_2^2 \leq 25 \quad (a)$$

by the Lagrange multiplier method.

Solution. The Lagrange function is

$$L(\mathbf{x}, u) = x_1 x_2 + u(x_1^2 + x_2^2 - 25) \quad (b)$$

The necessary conditions for a stationary point are

$$\frac{\partial L}{\partial x_1} = x_2 + 2ux_1 = 0$$

$$\frac{\partial L}{\partial x_2} = x_1 + 2ux_2 = 0$$

$$\frac{\partial L}{\partial u} = x_1^2 + x_2^2 \leq 25 \quad (c)$$

$$u(25 - x_1^2 - x_2^2) = 0$$

The five simultaneous solutions of Equations (c) are listed in Table E8.3. How would you calculate these values?

Columns two and three of Table E8.3 list the components of \mathbf{x}^* that are the stationary solutions of the problem. Note that the solutions with $u > 0$ are minima, those for $u < 0$ are maxima, and $u = 0$ is a saddle point. This is because maximizing f is equivalent to minimizing $-f$, and the KTC for the problem in Equation (a) with f replaced by $-f$ are the equations shown in (c) with u allowed to be negative. In Fig-

TABLE E8.3
Solutions of Example 8.3 by the Lagrange multiplier method

U	x_1	x_2	Point	C	$f(\mathbf{x})$	Remarks
0	0	0	A	25	0	saddle
0.5	+3.54	-3.54	B	0	-12.5	minimum
	-3.54	+3.54	C	0	-12.5	minimum
-0.5	+3.54	+3.54	D	0	+12.5	maximum
	-3.54	-3.54	E	0	+12.5	maximum

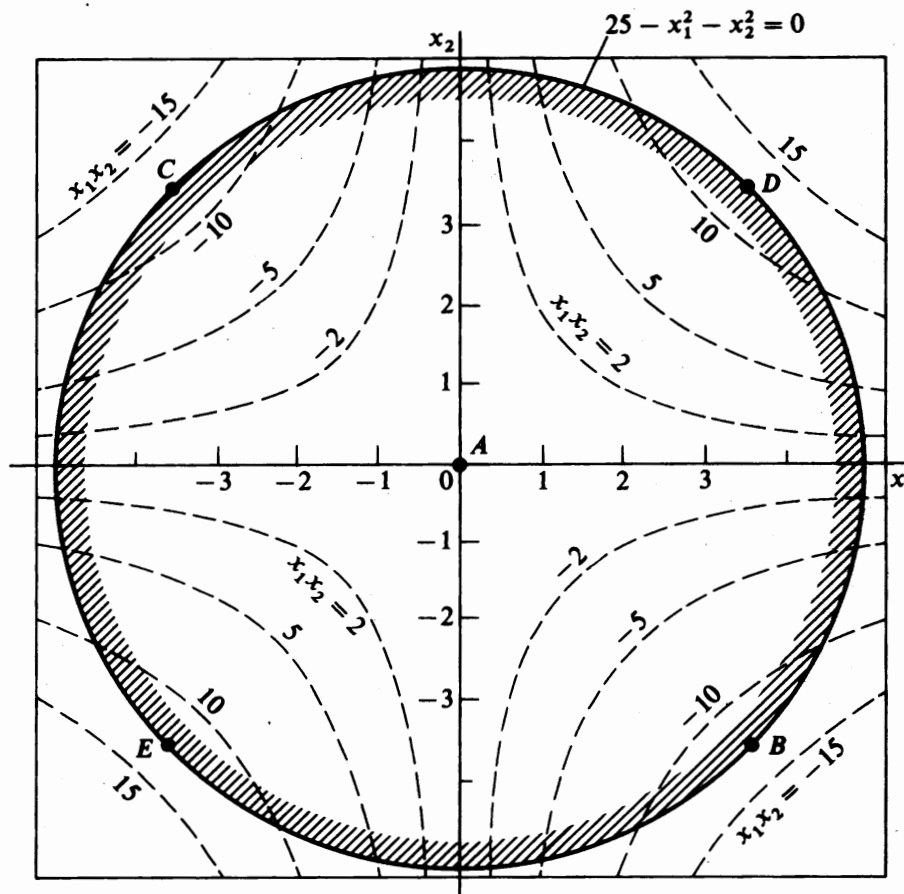


FIGURE E8.3

Figure E8.3 the contours of the objective function (hyperbolas) are represented by broken lines, and the feasible region is bounded by the shaded area enclosed by the circle $g(\mathbf{x}) = 25$. Points B and C correspond to the two minima, D and E to the two maxima, and A to the saddle point of $f(\mathbf{x})$.

Lagrange multipliers and sensitivity analysis

At each iteration, NLP algorithms form new estimates not only of the decision variables \mathbf{x} but also of the Lagrange multipliers $\boldsymbol{\lambda}$ and \mathbf{u} . If, at these estimates, all constraints are satisfied and the KTC are satisfied to within specified tolerances, the algorithm stops. At a local optimum, the optimal multiplier values provide useful sensitivity information. In the NLP (8.25)–(8.26), let $V^*(\mathbf{b}, \mathbf{c})$ be the optimal value of the objective f at a local minimum, viewed as a function of the right-hand sides of the constraints \mathbf{b} and \mathbf{c} . Then, under additional conditions (see Luenberger, 1984, Chapter 10)

$$\lambda_i^* = \frac{-\partial V^*}{\partial b_i}, \quad i = 1, \dots, m \quad (8.30a)$$

$$u_j^* = \frac{-\partial V^*}{\partial c_j}, \quad j = 1, \dots, r \quad (8.30b)$$

That is, the Lagrange multipliers provide the rate of change of the optimal objective value with respect to changes in the constraint right-hand sides. This information is often of significant value. For example, if the right-hand side of an inequality constraint c_j represents the capacity of a process and this capacity constraint is active at the optimum, then the optimal multiplier value u_j^* equals the rate of decrease of the minimal cost if the capacity is increased. This change is the marginal value of the capacity. In a situation with several active capacity limits, the ones with the largest absolute multipliers should be considered first for possible increases. Examples of the use of Lagrange multipliers for sensitivity analysis in linear programming are given in Chapter 7.

Lagrange multipliers are quite helpful in analyzing parameter sensitivities in problems with multiple constraints. In a typical refinery, a number of different products are manufactured which must usually meet (or exceed) certain specifications in terms of purity as required by the customers. Suppose we carry out a constrained optimization for an objective function that includes several variables that occur in the refinery model, that is, those in the fluid catalytic cracker, in the distillation column, and so on, and arrive at some economic optimum subject to the constraints on product purity. Given the optimum values of the variables plus the Lagrange multipliers corresponding to the product purity, we can then pose the question: How will the profits change if the product specification is either relaxed or made more stringent? To answer this question simply requires examining the Lagrange multiplier for each constraint. As an example, consider the case in which there are three major products (A , B , and C) and the Lagrange multipliers corresponding to each of the three demand inequality constraints are calculated to be:

$$u_A = -0.001$$

$$u_B = -1.0$$

$$u_C = -0.007$$

The values for u_i show (ignoring scaling) that satisfying an additional unit of demand of product B is much more costly than for the other two products.

Convex programming problems

The KTC comprise both the necessary and sufficient conditions for optimality for smooth convex problems. In the problem (8.25)–(8.26), if the objective $f(\mathbf{x})$ and inequality constraint functions g_j are convex, and the equality constraint functions h_j are linear, then the feasible region of the problem is convex, and any local minimum is a global minimum. Further, if \mathbf{x}^* is a feasible solution, if all the problem functions have continuous first derivatives at \mathbf{x}^* , and if the gradients of the active constraints at \mathbf{x}^* are independent, then \mathbf{x}^* is optimal if and only if the KTC are satisfied at \mathbf{x}^* .

Practical considerations

Many real problems do not satisfy these convexity assumptions. In chemical engineering applications, equality constraints often consist of input–output relations of process units that are often nonlinear. Convexity of the feasible region can only be guaranteed if these constraints are all linear. Also, it is often difficult to tell if an inequality constraint or objective function is convex or not. Hence it is often uncertain if a point satisfying the KTC is a local or global optimum, or even a saddle point. For problems with a few variables we can sometimes find all KTC solutions analytically and pick the one with the best objective function value. Otherwise, most numerical algorithms terminate when the KTC are satisfied to within some tolerance. The user usually specifies two separate tolerances: a feasibility tolerance ε_f and an optimality tolerance ε_o . A point \bar{x} is feasible to within ε_f if

$$|h_i(\bar{x}) - b_i| \leq \varepsilon_f, \quad \text{for } i = 1, \dots, m$$

and

$$g_j(\bar{x}) - c_j \leq \varepsilon_f, \quad \text{for } j = 1, \dots, r \quad (8.31a)$$

Furthermore, \bar{x} is optimal to within $(\varepsilon_o, \varepsilon_f)$ if it is feasible to within ε_f and the KTC are satisfied to within ε_o . This means that, in Equations (8.23)–(8.24)

$$\left| \frac{\partial L}{\partial x_i}(\bar{x}, \bar{\lambda}, \bar{u}) \right| \leq \varepsilon_o, \quad i = 1, \dots, n$$

and

$$u_j \geq -\varepsilon_o, \quad j = 1, \dots, r \quad (8.31b)$$

Equation (8.31b) corresponds to relaxing the constraint.

Second-order necessary and sufficiency conditions for optimality

The Kuhn–Tucker necessary conditions are satisfied at any local minimum or maximum and at saddle points. If $(\mathbf{x}^*, \boldsymbol{\lambda}^*, \mathbf{u}^*)$ is a Kuhn–Tucker point for the problem (8.25)–(8.26), and the second-order sufficiency conditions are satisfied at that point, optimality is guaranteed. The second order optimality conditions involve the matrix of second partial derivatives with respect to \mathbf{x} (the Hessian matrix of the Lagrangian function), and may be written as follows:

$$\mathbf{y}^T \nabla_{\mathbf{x}}^2 L(\mathbf{x}^*, \boldsymbol{\lambda}^*, \mathbf{u}^*) \mathbf{y} > 0 \quad (8.32a)$$

for all nonzero vectors \mathbf{y} such that

$$\mathbf{J}(\mathbf{x}^*) \mathbf{y} = \mathbf{0} \quad (8.32b)$$

where $\mathbf{J}(\mathbf{x}^*)$ is the matrix whose rows are the gradients of the constraints that are active at \mathbf{x}^* . Equation (8.32b) defines a set of vectors \mathbf{y} that are orthogonal to the gradients of the active constraints. These vectors constitute the *tangent plane* to the

active constraints, which was illustrated in Example 8.1. Hence (8.32a) requires that the Lagrangian Hessian matrix be positive-definite for all vectors \mathbf{y} on this tangent plane. If the “ $>$ ” sign in (8.32a) is replaced by “ \geq ”, then (8.32a)–(8.32b) plus the KTC are the *second-order necessary conditions* for a local minimum. See Luenberger (1984) or Nash and Sofer (1996) for a more thorough discussion of these second-order conditions.

If no active constraints occur (so \mathbf{x}^* is an unconstrained stationary point), then (8.32a) must hold for all vectors \mathbf{y} , and the multipliers $\boldsymbol{\lambda}^*$ and \mathbf{u}^* are zero, so $\nabla_{\mathbf{x}}^2 L = \nabla_{\mathbf{x}}^2 f$. Hence (8.32a) and (8.32b) reduce to the condition discussed in Section 4.5 that if the Hessian matrix of the objective function, evaluated at \mathbf{x}^* , is positive-definite and \mathbf{x}^* is a stationary point, then \mathbf{x}^* is a local unconstrained minimum of f .

EXAMPLE 8.4 USING THE SECOND-ORDER CONDITIONS

As an example, consider the problem:

$$\text{Minimize: } f(\mathbf{x}) = (x_1 - 1)^2 + x_2^2$$

$$\text{Subject to: } x_1 - x_2^2 \leq 0$$

Solution. Although the objective function of this problem is convex, the inequality constraint does not define a convex feasible region; as shown in Figure E8.4. The geometric interpretation is to find the points in the feasible region closest to $(1, 0)$. The Lagrangian function for this problem is

$$L(\mathbf{x}, u) = (x_1 - 1)^2 + x_2^2 + u(x_1 - x_2^2)$$

and the KTC for a local minimum are

$$\partial L / \partial x_1 = 2(x_1 - 1) + u = 0$$

$$\partial L / \partial x_2 = 2x_2 - 2ux_2 = 0$$

$$x_1 - x_2^2 \leq 0, \quad u \geq 0$$

There are three solutions to these conditions: two global minima, at $x_1^* = \frac{1}{2}$, $x_2^* = \pm\sqrt{\frac{1}{2}}$, $u^* = 1$ with an objective value of 0.75, and a local maximum at $x_1^0 = 0$, $x_2^0 = 0$, $u^0 = 2$ with an objective value of 1.0. These solutions are evident by examining Figure E8.4.

The second order sufficiency conditions show that the first two of these three Kuhn–Tucker points are local minima, and the third is not. The Hessian matrix of the Lagrangian function is

$$\nabla_{\mathbf{x}}^2 L(x, u) = \begin{bmatrix} 2 & 0 \\ 0 & 2(1 - u) \end{bmatrix}$$

The Hessian evaluated at $(x_1^0 = 0, x_2^0 = 0, u^0 = 2)$ is

$$\nabla_{\mathbf{x}}^2 L(0, 0, 2) = \begin{bmatrix} 2 & 0 \\ 0 & -2 \end{bmatrix}$$

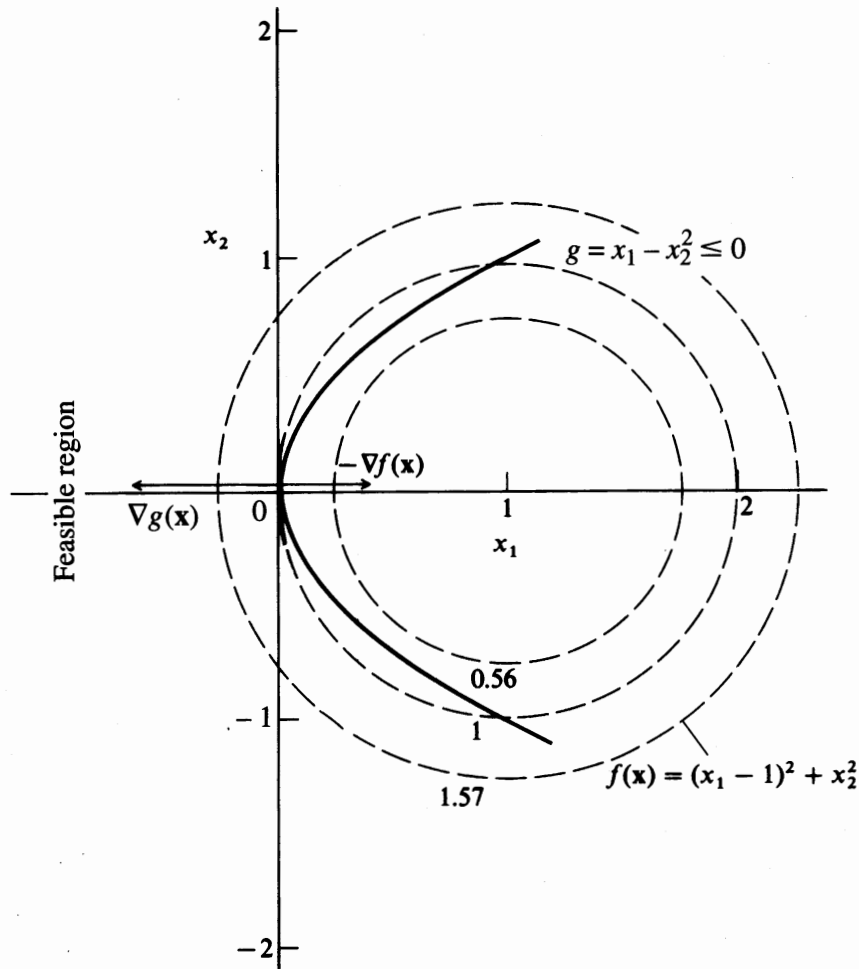


FIGURE E8.4

The second-order necessary conditions require this matrix to be positive-semidefinite on the tangent plane to the active constraints at $(0, 0)$, as defined in expression (8.32b). Here, this tangent plane is the set

$$T = \{\mathbf{y} \mid \nabla g(0,0)^T \mathbf{y} = 0\}$$

The gradient of the constraint function is

$$\nabla^T g(x_1, x_2) = [1 \quad -2x_2] \quad \text{so} \quad \nabla g(0, 0) = [1 \quad 0]$$

Thus the tangent plane at $(0, 0)$ is

$$T = \{\mathbf{y} \mid y_1 = 0\} = \{\mathbf{y} \mid \mathbf{y} = (0, y_2)\}$$

and the quadratic form in (8.32a), evaluated on the tangent plane, is

$$\mathbf{y}^T \nabla_x^2 L(\mathbf{x}^*, \mathbf{u}^*) \mathbf{y} = [0 \quad y_2] \begin{bmatrix} 2 & 0 \\ 0 & -2 \end{bmatrix} \begin{bmatrix} 0 \\ y_2 \end{bmatrix} = -2y_2^2$$

Because $-2y_2^2$ is negative for all nonzero vectors in the set T , the second-order necessary condition is not satisfied, so $(0, 0)$ is not a local minimum.

If we check the minimum at $x_1^* = \frac{1}{2}, x_2^* = \sqrt{\frac{1}{2}}, u^* = 1$, the Lagrangian Hessian evaluated at this point is

$$\nabla_x^2 L\left(\frac{1}{2}, \sqrt{\frac{1}{2}}, 1\right) = \begin{bmatrix} 2 & 0 \\ 0 & 0 \end{bmatrix}$$

The constraint gradient at this point is $[1 \quad -\sqrt{2}]$, so the tangent plane is

$$T = \{\mathbf{y} | y_1 - \sqrt{2}y_2 = 0\} = \{\mathbf{y} | y = y_2(-\sqrt{2}, 1)\}$$

On this tangent plane, the quadratic form is

$$\mathbf{y}^T \nabla_x^2 L(\mathbf{x}^*, \mathbf{u}^*) \mathbf{y} = y_2^2 \begin{bmatrix} -\sqrt{2} & 1 \end{bmatrix} \begin{bmatrix} 2 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} -\sqrt{2} \\ 1 \end{bmatrix} = 4y_2^2$$

This is positive for all nonzero vectors in the set T , so the second-order sufficiency conditions are satisfied, and the point is a local minimum.

8.3 QUADRATIC PROGRAMMING

A *quadratic programming* (QP) problem is an optimization problem in which a quadratic objective function of n variables is minimized subject to m linear inequality or equality constraints. A convex QP is the simplest form of a nonlinear programming problem with inequality constraints. A number of practical optimization problems, such as constrained least squares and optimal control of linear systems with quadratic cost functions and linear constraints, are naturally posed as QP problems. In this text we discuss QP as a subproblem to solve general nonlinear programming problems. The algorithms used to solve QPs bear many similarities to algorithms used in solving the linear programming problems discussed in Chapter 7.

In matrix notation, the quadratic programming problem is

$$\begin{aligned} \text{Minimize: } & f(\mathbf{x}) = \mathbf{c}^T \mathbf{x} + \frac{1}{2} \mathbf{x}^T \mathbf{Q} \mathbf{x} \\ \text{Subject to: } & \mathbf{A} \mathbf{x} = \mathbf{b} \\ & \mathbf{x} \geq \mathbf{0} \end{aligned} \tag{8.33}$$

where \mathbf{c} is a vector of constant coefficients, \mathbf{A} is an $(m \times n)$ matrix, and \mathbf{Q} is a symmetric matrix.

The vector \mathbf{x} can contain slack variables, so the equality constraints (8.33) may contain some constraints that were originally inequalities but have been converted to equalities by inserting slacks. Codes for quadratic programming allow arbitrary upper and lower bounds on \mathbf{x} ; we assume $\mathbf{x} \geq \mathbf{0}$ only for simplicity.

If the equality constraints in (8.33) are independent then, as discussed in Section 8.2, the KTC are the necessary conditions for an optimal solution of the QP. In addition, if \mathbf{Q} is positive-semidefinite in (8.33), the QP objective function is con-

vex. Because the feasible region of a QP is defined by linear constraints, it is always convex, so the QP is then a convex programming problem, and any local solution is a global solution. Also, the KTC are the sufficient conditions for a minimum, and a solution meeting these conditions yields the global optimum. If \mathbf{Q} is not positive-semidefinite, the problem may have an unbounded solution or local minima.

To write the KTC, start with the Lagrangian function

$$L = \mathbf{x}^T \mathbf{c} + \frac{1}{2} \mathbf{x}^T \mathbf{Q} \mathbf{x} + \boldsymbol{\lambda}^T (\mathbf{A} \mathbf{x} - \mathbf{b}) - \mathbf{u}^T \mathbf{x}$$

and equate the gradient of L (with respect to \mathbf{x}^T) to zero (note that $\boldsymbol{\lambda}^T (\mathbf{A} \mathbf{x} - \mathbf{b}) = (\mathbf{A} \mathbf{x} - \mathbf{b})^T \boldsymbol{\lambda} = (\mathbf{x}^T \mathbf{A}^T - \mathbf{b}^T) \boldsymbol{\lambda}$ and $\mathbf{u}^T \mathbf{x} = \mathbf{x}^T \mathbf{u}$)

$$\nabla_{\mathbf{x}} L = \mathbf{c} + \mathbf{Q} \mathbf{x} + \mathbf{A}^T \boldsymbol{\lambda} - \mathbf{u} = \mathbf{0}$$

Then the KTC reduce to the following set of equations:

$$\mathbf{c} + \mathbf{Q} \mathbf{x} + \mathbf{A}^T \boldsymbol{\lambda} - \mathbf{u} = \mathbf{0} \quad (8.34)$$

$$\mathbf{A} \mathbf{x} - \mathbf{b} = \mathbf{0} \quad (8.35)$$

$$\mathbf{x} \geq \mathbf{0} \quad \mathbf{u} \geq \mathbf{0} \quad (8.36)$$

$$\mathbf{u}^T \mathbf{x} = 0 \quad (8.37)$$

where the u_i and λ_j are the Lagrange multipliers. If \mathbf{Q} is positive semidefinite, any set of variables $(\mathbf{x}^*, \mathbf{u}^*, \boldsymbol{\lambda}^*)$ that satisfies (8.34) to (8.37) is an optimal solution to (8.33).

Some QP solvers use these KTC directly by finding a solution satisfying the equations. They are linear except for (8.37), which is called a complementary slackness condition. These conditions were discussed for general inequality constraints in Section 8.2. Applied to the nonnegativity conditions in (8.33), complementary slackness implies that at least one of each pair of variables (u_i, x_i) must be zero. Hence a feasible solution to the KTC can be found by starting with an infeasible complementary solution to the linear constraints (8.34)–(8.36) and using LP pivot operations to minimize the sum of infeasibilities while maintaining complementarity. Because (8.34) and (8.35) have n and m constraints, respectively, the effect is roughly equivalent to solving an LP with $(n + m)$ rows. Because LP “machinery” is used, most commercial LP systems, including those discussed in Chapter 7, contain QP solvers. In addition, a QP can also be solved by any efficient general purpose NLP solver.

8.4 PENALTY, BARRIER, AND AUGMENTED LAGRANGIAN METHODS

The essential idea of a penalty method of nonlinear programming is to transform a constrained problem into a sequence of unconstrained problems.

$$\left. \begin{array}{l} \text{Minimize: } f(\mathbf{x}) \\ \text{Subject to: } \mathbf{g}(\mathbf{x}) \leq \mathbf{0} \\ \mathbf{h}(\mathbf{x}) = \mathbf{0} \end{array} \right\} \Rightarrow \text{Minimize: } P(f, \mathbf{g}, \mathbf{h}, r) \quad (8.38)$$

where $P(f, \mathbf{g}, \mathbf{h}, r)$ is a *penalty function*, and r is a positive penalty parameter. After the penalty function is formulated, it is minimized for a series of values of increasing r -values, which force the sequence of minima to approach the optimum of the constrained problem.

As an example, consider the problem

$$\text{Minimize: } f(\mathbf{x}) = (x_1 - 1)^2 + (x_2 - 2)^2$$

$$\text{Subject to: } h(\mathbf{x}) = x_1 + x_2 - 4 = 0$$

We formulate a new unconstrained objective function

$$P(\mathbf{x}, r) = (x_1 - 1)^2 + (x_2 - 2)^2 + r(x_1 + x_2 - 4)^2$$

where r is a positive scalar called the penalty parameter, and $r(x_1 + x_2 - 4)^2$ is called the penalty term. Consider a series of minimization problems where we minimize $P(\mathbf{x}, r)$ for an increasing sequence of r values tending to infinity. As r increases, the penalty term becomes large for any values of \mathbf{x} that violate the equality constraints in (8.38). As the penalty term grows, the values of x_i change to those that cause the equality constraint to be satisfied. In the limit the product of r and h^2 approaches zero so that the value of f approaches the value of P . This is shown in Figure 8.5. The constrained optimum is $\mathbf{x}^* = (1.5, 2.5)$ and the unconstrained minimum of the objective is at $(1, 2)$. The point $(1, 2)$ is also the minimum of $P(\mathbf{x}, 0)$. The minimizing points for $r = 1, 10, 100, 1000$ are at the center of the elliptical contours in the figure. Table 8.1 shows $r, x_1(r),$ and $x_2(r)$. It is clear that $\mathbf{x}(r) \rightarrow \mathbf{x}^*$ as $r \rightarrow \infty$, which can be shown to be true in general (see Luenberger, 1984).

Note how the contours of $P(\mathbf{x}, r)$ bunch up around the constraint line $x_1 + x_2 = 4$ as r becomes large. This happens because, for large r , $P(\mathbf{x}, r)$ increases rapidly as violations of $x_1 + x_2 = 4$ increase, that is, as you move away from this line. This bunching and elongation of the contours of $P(\mathbf{x}, r)$ shows itself in the condition number of $\nabla^2 P(\mathbf{x}, r)$, the Hessian matrix of P . As shown in Appendix A, the condition number of a positive-definite matrix is the ratio of the largest to smallest eigenvalue. Because for large values of r , the eigenvalue ratio is large, $\nabla^2 P$ is said to be *ill-conditioned*. In fact, the condition number of $\nabla^2 P$ approaches ∞ as $r \rightarrow \infty$ (see Luenberger, 1984), so P becomes harder and harder to minimize accurately.

TABLE 8.1
Effect of penalty weighting
coefficient r on minimum of f

r	x_1	x_2	f
0	1.0000	2.0000	0.0000
0.1	1.0833	2.0833	0.0833
1	1.3333	2.3333	0.3333
10	1.4762	2.4762	0.4762
100	1.4975	2.4975	0.4975
1000	1.4998	2.4998	0.4998
\mathbf{x}^*	1.5000	2.5000	0.5000

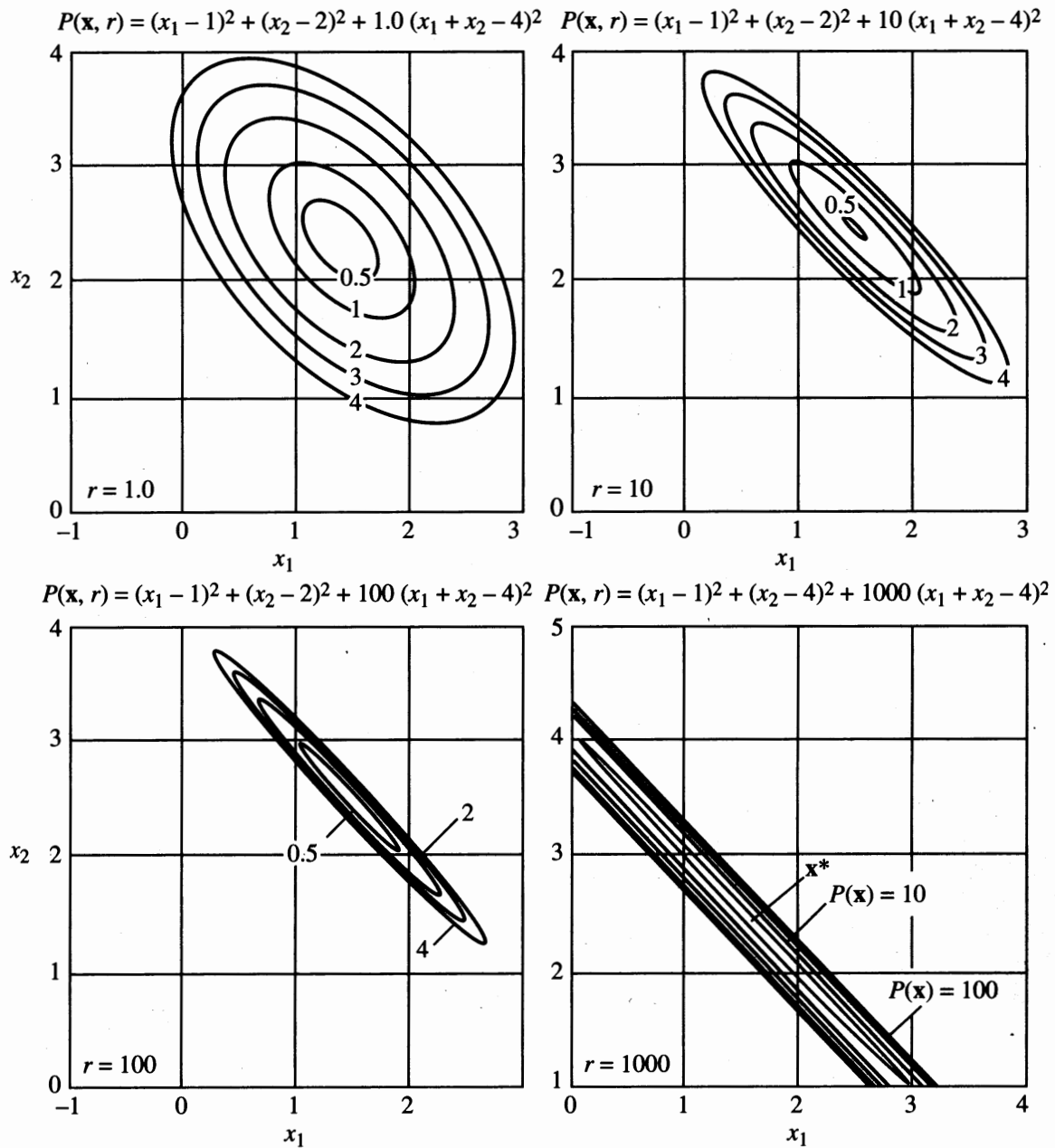


FIGURE 8.5

Transformation of a constrained problem to an unconstrained equivalent problem. The contours of the unconstrained penalty function are shown for different values of r .

The condition number of the Hessian matrix of the objective function is an important measure of difficulty in unconstrained optimization. By definition, the smallest a condition number can be is 1.0. A condition number of 10^5 is moderately large, 10^9 is large, and 10^{14} is extremely large. Recall that, if Newton's method is used to minimize a function f , the Newton search direction \mathbf{s} is found by solving the linear equations

$$(\nabla^2 f)\mathbf{s} = -\nabla f$$

These equations become harder and harder to solve numerically as $\nabla^2 f$ becomes more ill-conditioned. When its condition number exceeds 10^{14} , there will be few if any correct digits in the computed solution using double precision arithmetic (see Luenberger, 1984).

Because of the occurrence of ill-conditioning, “pure” penalty methods have been replaced by more efficient algorithms. In SLP and SQP, a “merit function” is used within the line search phase of these algorithms.

The general form of the quadratic penalty function for a problem of the form (8.25)–(8.26) with both equality and inequality constraints is

$$P_2(\mathbf{x}, r) = f(\mathbf{x}) + r \left(\sum_{j=1}^m h_j^2(\mathbf{x}) + \sum_{j=1}^r [\max\{0, g_j(\mathbf{x})\}]^2 \right) \quad (8.39)$$

The maximum-squared term ensures that a positive penalty is incurred only when the $g_j \leq 0$ constraint is violated.

An exact penalty function

Consider the exact L_1 penalty function; The term “ L_1 ” means that the L_1 (absolute value) norm is used to measure infeasibilities.

$$P_1(\mathbf{x}, w_1, w_2) = f(\mathbf{x}) + \left[\sum_{j=1}^m w_1 |h_j(\mathbf{x})| + \sum_{j=1}^r w_2 \max\{0, g_j(\mathbf{x})\} \right] \quad (8.40)$$

where the w_1 and w_2 are positive weights. The second term in h_j produces the same effect as the squared terms in Equation (8.39). When a constraint is violated, there is a positive contribution to the penalty term equal to the amount of the violation rather than the squared amount. In fact, this “sum of violations” or *sum of infeasibilities* is the objective used in phase one of the simplex method to find a feasible solution to a linear program (see Chapter 7).

Let \mathbf{x}^* be a local minimum of the problem (8.25)–(8.26), and let $(\boldsymbol{\lambda}^*, \mathbf{u}^*)$ be a vector of optimal multipliers corresponding to \mathbf{x}^* , that is, $(\mathbf{x}^*, \boldsymbol{\lambda}^*, \mathbf{u}^*)$ satisfy the KTC (8.27)–(8.29). If

$$w_1 \geq |\lambda_j^*|, \quad j = 1, \dots, m \quad (8.41)$$

$$w_2 \geq |u_j^*|, \quad j = 1, \dots, r \quad (8.42)$$

then \mathbf{x}^* is a local minimum of $P_1(\mathbf{x}, w_1, w_2)$. For a proof, see Luenberger (1984). If each penalty weight is larger than the absolute value of the corresponding optimal multiplier, the constrained problem can be solved by a *single* unconstrained minimization of P_1 . The penalty weights do not have to approach $+\infty$, and no infinite ill-conditioning occurs. This is why P_1 is called “exact.” There are other exact penalty functions; for example, the “augmented Lagrangian” will be discussed subsequently.

Intuitively, P_1 is exact and the squared penalty function P_2 is not because squaring a small infeasibility makes it much smaller, that is, $(10^{-4})^2 = 10^{-8}$. Hence the penalty parameter r in P_2 must increase faster as the infeasibilities get small, and it can never be large enough to make all infeasibilities vanish.

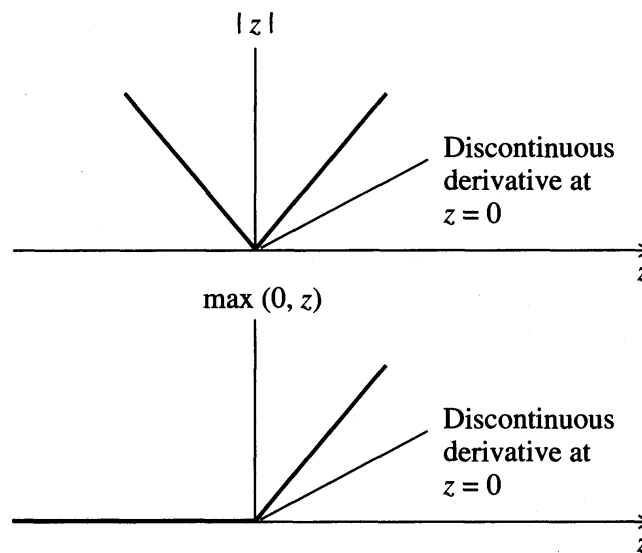


FIGURE 8.6
Discontinuous derivatives in the P_1 penalty function.

Despite the “exactness” feature of P_1 , no general-purpose, widely available NLP solver is based solely on the L_1 exact penalty function P_1 . This is because P_1 also has a negative characteristic; it is nonsmooth. The term $|h_j(\mathbf{x})|$ has a discontinuous derivative at any point \mathbf{x} where $h_j(\mathbf{x}) = 0$, that is, at any point satisfying the j th equality constraint; in addition, $\max\{0, g_j(\mathbf{x})\}$ has a discontinuous derivative at any \mathbf{x} where $g_j(\mathbf{x}) = 0$, that is, whenever the j th inequality constraint is active, as illustrated in Figure 8.6. These discontinuities occur at any feasible or partially feasible point, so none of the efficient unconstrained minimizers for smooth problems considered in Chapter 6 can be applied, because they eventually encounter points where P_1 is nonsmooth.

An equivalent smooth constrained problem

The problem of minimizing P_1 subject to no constraints is equivalent to the following smooth constrained problem.

$$\text{Minimize: } f(\mathbf{x}) + \sum_{j=1}^m w1_j(p1_j + n1_j) + \sum_{j=1}^r w2_j(p2_j) \quad (8.43)$$

$$\text{Subject to: } h_j(\mathbf{x}) = p1_j - n1_j, \quad j = 1, \dots, m \quad (8.44)$$

$$g_j(\mathbf{x}) = p2_j - n2_j, \quad j = 1, \dots, r \quad (8.45)$$

$$\text{all } p1_j, p2_j, n1_j, n2_j \geq 0 \quad (8.46)$$

The p 's are “positive deviation” variables and the n 's “negative deviation” variables. $p1_j$ and $p2_j$ equal h_j and g_j , respectively, when h_j and g_j are positive, and $n1_j$

and $n2_j$ equal h_j and g_j , respectively, when h_j and g_j are negative, providing that at most one variable in each pair $(p1_j, n1_j)$ and $(p2_j, u2_j)$ is positive, that is,

$$p1_j n1_j = 0, p2_j n2_j = 0 \quad (8.47)$$

But Equation (8.47) must hold at any optimal solution of (8.43)–(8.46), as long as all weights $w1_j$ and $w2_j$ are positive. To see why, consider the example $h_1 = -3$, $p1_1 = 2$, $n1_1 = 5$. The objective (8.43) contains a term $w1_1 (p1_1 + n1_1) = 7w1_1$. The new solution $p1_1 = 0$, $n1_1 = 3$ has an objective contribution of $5w1_1$, so the old solution cannot be optimal.

When (8.44)–(8.47) hold,

$$p1_j + n1_j = |h_j(\mathbf{x})|$$

and

$$p2_j = \max(0, g_j(\mathbf{x}))$$

so the objective (8.43) equals the L_1 exact penalty function (8.40).

The problem (8.43)–(8.46) is called an “elastic” formulation of the original “inelastic” problem (8.11), because the deviation variables allow the constraints to “stretch” (i.e., be violated) at costs per unit of violation $w1_j$ and $w2_j$. This idea of allowing constraints to be violated, but at a price, is an important modeling concept that is widely used. Constraints expressing physical laws or “hard” limits cannot be treated this way—this is equivalent to using infinite weights. However many other constraints are really “soft,” for example some customer demands and capacity limits. For further discussions of elastic programming, see Brown (1997). Curve-fitting problems using absolute value (L_1) or minimax (L_∞) norms can also be formulated as smooth constrained problems using deviation variables, as can problems involving multiple objectives, using “goal programming” (Rustem, 1998).

Augmented Lagrangians

The “augmented Lagrangian” is a smooth exact penalty function. For simplicity, we describe it for problems having only equality constraints, but it is easily extended to problems that include inequalities. The augmented Lagrangian function is

$$AL(\mathbf{x}, \boldsymbol{\lambda}, r) = f(\mathbf{x}) + \sum_{j=1}^m \lambda_j h_j(\mathbf{x}) + r \sum_{j=1}^m h_j^2(\mathbf{x}) \quad (8.48)$$

where r is a positive penalty parameter, and the λ_j are Lagrange multipliers. AL is simply the Lagrangian L plus a squared penalty term. Let \mathbf{x}^* be a local minimum of the equality constrained problem

$$\text{Minimize: } f(\mathbf{x})$$

$$\text{Subject to: } h_j(\mathbf{x}) = 0, \quad j = 1, \dots, m$$

and let $(\mathbf{x}^*, \boldsymbol{\lambda}^*)$ satisfy the KTC for this problem. The gradient of AL is

$$\nabla_{\mathbf{x}} AL(\mathbf{x}, \boldsymbol{\lambda}, r) = \nabla f(\mathbf{x}) + \sum_{j=1}^m \lambda_j \nabla h_j(\mathbf{x}) + 2r \sum_{j=1}^m h_j(\mathbf{x}) \nabla h_j(\mathbf{x}) \quad (8.49)$$

Since \mathbf{x}^* is feasible, $h_j(\mathbf{x}^*) = 0$, so if $\boldsymbol{\lambda}$ is set to $\boldsymbol{\lambda}^*$ in the augmented Lagrangian,

$$\nabla_{\mathbf{x}} AL(\mathbf{x}^*, \boldsymbol{\lambda}^*, r) = \nabla f(\mathbf{x}^*) + \sum_{j=1}^m \lambda_j^* \nabla h_j(\mathbf{x}^*) = 0 \quad (8.50)$$

Hence \mathbf{x}^* is a stationary point of $AL(\mathbf{x}, \boldsymbol{\lambda}^*, r)$ for any r . Not all stationary points are minima, but if $\nabla_{\mathbf{x}}^2 AL(\mathbf{x}^*, \boldsymbol{\lambda}^*, r)$ is positive-definite, then \mathbf{x}^* satisfies the second-order sufficiency conditions, and so it is a local minimum. Luenberger (1984) shows that this is true if r is large enough, that is, there is a threshold $\bar{r} > 0$ such that, if $r > \bar{r}$, then $\nabla_{\mathbf{x}}^2 AL(\mathbf{x}^*, \boldsymbol{\lambda}^*, r)$ is positive-definite. Hence for $r > \bar{r}$, $AL(\mathbf{x}, \boldsymbol{\lambda}^*, r)$ is an exact penalty function.

Again, there is a “catch.” In general, \bar{r} and $\boldsymbol{\lambda}^*$ are unknown. Algorithms have been developed that perform a sequence of minimizations of AL , generating successively better estimates of $\boldsymbol{\lambda}^*$ and increasing r if necessary [see Luenberger (1984)]. However, NLP solvers based on these algorithms have now been replaced with better ones based on the SLP, SQP, or GRG algorithms described in this chapter. The function AL does, however, serve as a line search objective in some SQP implementations; see Nocedal and Wright (1999).

Barrier methods

Like penalty methods, *barrier methods* convert a constrained optimization problem into a series of unconstrained ones. The optimal solutions to these unconstrained subproblems are in the interior of the feasible region, and they converge to the constrained solution as a positive barrier parameter approaches zero. This approach contrasts with the behavior of penalty methods, whose unconstrained subproblem solutions converge from outside the feasible region.

To illustrate, consider the example used at the start of Section 8.4 to illustrate penalty methods, but with the equality constraint changed to an inequality:

$$\text{Minimize: } f(\mathbf{x}) = (x_1 - 1)^2 + (x_2 - 2)^2$$

$$\text{Subject to: } g(\mathbf{x}) = x_1 + x_2 - 4 \geq 0$$

The equality constrained problem was graphed in Figure 8.5. The feasible region is now the set of points on and above the line $x_1 + x_2 - 4 = 0$, and the constrained solution is still at the point (1.5, 2.5) where $f = 0.5$.

The logarithmic barrier function for this problem is

$$\begin{aligned} B(\mathbf{x}, r) &= f(\mathbf{x}) - r \ln(g(\mathbf{x})) \\ &= (x_1 - 1)^2 + (x_2 - 2)^2 - r \ln(x_1 + x_2 - 4) \end{aligned}$$

where r is a positive scalar called the barrier parameter. This function is defined only in the interior of the feasible region, where $g(\mathbf{x})$ is positive. Consider minimizing B starting from an interior point. As \mathbf{x} approaches the constraint boundary, $g(\mathbf{x})$ approaches zero, and the barrier term $-r \ln(g(\mathbf{x}))$ approaches infinity, so it creates an infinitely high barrier along this boundary. The penalty forces B to have an

TABLE 8.2
Convergence of barrier function $B(\mathbf{x}, r)$

Barrier parameter, r	$x_1(r)$	$x_2(r)$	Objective	Value of the constraint	Barrier term	$B(\mathbf{x}, r)$
10	2.851	3.851	6.851	2.702	9.938	-3.088
5	2.396	3.396	3.896	1.791	2.915	0.981
1	1.809	2.809	1.309	0.618	-0.481	1.790
0.1	1.546	2.546	0.596	0.092	-0.239	0.835
0.01	1.505	2.505	0.510	0.010	-0.046	0.556
\mathbf{x}^*	1.500	2.500	0.500	0.000	0.000	0.500

unconstrained minimum in the interior of the feasible region, and its location depends on the barrier parameter r . If $\mathbf{x}(r)$ is an unconstrained interior minimum of $B(\mathbf{x}, r)$, then as r approaches zero, the barrier term has a decreasing weight, so $\mathbf{x}(r)$ can approach the boundary of the feasible region if the constrained solution is on the boundary. As r approaches zero, $\mathbf{x}(r)$ approaches an optimal solution of the original problem, as shown in Nash and Sofer (1996) and Nocedal and Wright (1999).

To illustrate this behavior, Table 8.2 shows the optimal unconstrained solutions and their associated objective, constraint, and barrier function values for the preceding problem, for a sequence of decreasing r values.

For larger r values, $\mathbf{x}(r)$ is forced further from the constraint boundary. In contrast, as r approaches zero, $x_1(r)$ and $x_2(r)$ converge to their optimal values of 1.5 and 2.5, respectively, and the constraint value approaches zero. The term $-\ln(g(\mathbf{x}))$ approaches infinity, but the weighted barrier term $-r \ln(g(\mathbf{x}))$ approaches zero, and the value of B approaches the optimal objective value.

For a general problem with only inequality constraints:

$$\text{Minimize: } f(\mathbf{x})$$

$$\text{Subject to: } g_i(\mathbf{x}) \geq 0, \quad i = 1, \dots, m$$

the logarithmic barrier function formulation is

$$\text{Minimize: } B(\mathbf{x}, r) = f(\mathbf{x}) - r \sum_{i=1}^m \ln(g_i(\mathbf{x}))$$

As with penalty functions, the condition number of the Hessian matrix $\nabla_{\mathbf{x}}^2 B(\mathbf{x}(r), r)$ approaches infinity as r approaches zero, so B is very difficult to minimize accurately for small r . From a geometric viewpoint, this is because the barrier term approaches infinity rapidly as you move toward the boundary of the feasible region, so the contours of B “bunch up” near this boundary. Hence the barrier approach is not widely used today as a direct method of solving nonlinear programs. When a logarithmic barrier term is used to incorporate only the bounds on the variables, however, this leads to a barrier or *interior-point* method. This approach is very successful in solving large linear programs and is very promising for NLP problems as well. See Nash and Sofer (1996) or Nocedal and Wright (1999) for further details.

Barrier methods are not directly applicable to problems with equality constraints, but equality constraints can be incorporated using a penalty term and inequalities can use a barrier term, leading to a “mixed” penalty–barrier method.

8.5 SUCCESSIVE LINEAR PROGRAMMING

Successive linear programming (SLP) methods solve a sequence of linear programming approximations to a nonlinear programming problem. Recall that if $g_i(\mathbf{x})$ is a nonlinear function and \mathbf{x}^0 is the initial value for \mathbf{x} , then the first two terms in the Taylor series expansion of $g_i(\mathbf{x})$ around \mathbf{x}^0 are

$$g_i(\mathbf{x}) = g_i(\mathbf{x}^0 + \Delta\mathbf{x}) \cong g_i(\mathbf{x}^0) + \nabla g_i(\mathbf{x}^0)^T(\Delta\mathbf{x})$$

The error in this linear approximation approaches zero proportionally to $(\Delta\mathbf{x})^2$ as $\Delta\mathbf{x}$ approaches zero. Given initial values for the variables, all nonlinear functions in the problem are linearized and replaced by their linear Taylor series approximations at this initial point. The variables in the resulting LP are the $\Delta\mathbf{x}_i$'s, representing changes from the base values. In addition, upper and lower bounds (called *step bounds*) are imposed on these change variables because the linear approximation is reasonably accurate only in some neighborhood of the initial point.

The resulting LP is solved; if the new point is an improvement, it becomes the current point and the process is repeated. If the new point does not represent an improvement in the objective, we may be close enough to the optimum to stop or the step bounds may need to be reduced. Successive points generated by this procedure need not be feasible even if the initial point is. The extent of infeasibility generally is reduced as the iterations proceed, however.

We illustrate the basic concepts with a simple example. Consider the following problem:

$$\begin{aligned} \text{Maximize: } & 2x + y \\ \text{Subject to: } & x^2 + y^2 \leq 25 \\ & x^2 - y^2 \leq 7 \end{aligned}$$

and

$$\begin{aligned} x & \geq 0 \\ y & \geq 0 \end{aligned}$$

with an initial starting point of $(x_c, y_c) = (2, 2)$. Figure 8.7 shows the two nonlinear constraints and one objective function contour with an objective value of 10. Because the value of the objective function increases with increasing x and y , the figure shows that the optimal solution is at the point where the two nonlinear inequalities $x^2 + y^2 \leq 25$ and $x^2 - y^2 \leq 7$ are active, that is, at the solution of $x^2 + y^2 = 25$ and $x^2 - y^2 = 7$, which is $\mathbf{x}^* = (4, 3)$.

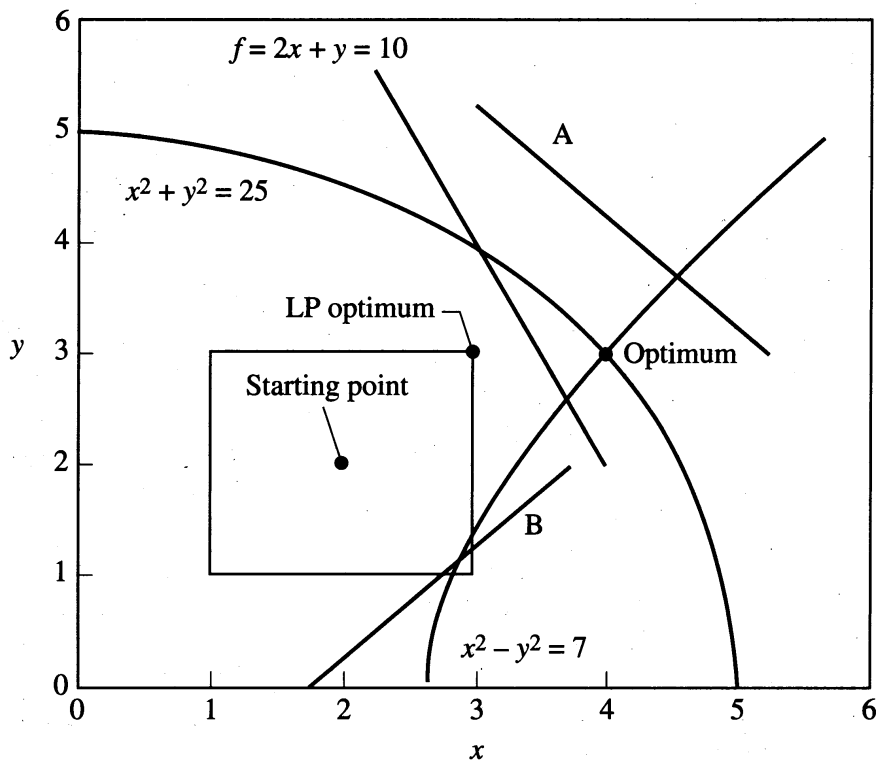


FIGURE 8.7

SLP example with linear objective, nonlinear constraints. Line A is the linearization of $x^2 + y^2 \leq 25$ and line B is the linearization of $x^2 - y^2 \leq 7$.

Next consider any optimization problem with n variables. Let \bar{x} be any feasible point, and let $n_{\text{act}}(\bar{x})$ be the number of active constraints at \bar{x} . Recall that a constraint is active at \bar{x} if it holds as an equality constraint there. Hence all equality constraints are active at any feasible point, but an inequality constraint may be active or inactive. Remember to include simple upper or lower bounds on the variables when counting active constraints. We define the number of degrees of freedom at \bar{x} as

$$\text{dof}(\bar{x}) = n - n_{\text{act}}(\bar{x})$$

Definition: A feasible point \bar{x} is called a *vertex* if $\text{dof}(\bar{x}) \leq 0$, and the Jacobian of the active constraints at \bar{x} has rank n where n is the number of variables. It is a *nondegenerate vertex* if $\text{dof}(\bar{x}) = 0$, and a *degenerate vertex* if $\text{dof}(\bar{x}) < 0$, in which case $|\text{dof}(\bar{x})|$ is called the *degree of degeneracy* at \bar{x} .

The requirement that there be at least n independent linearized constraints at \bar{x} is included to rule out situations where, for example, some of the active constraints are just multiples of one another. In the example $\text{dof}(\bar{x}) = 0$.

Returning to the example, the optimal point $x^* = (4, 3)$ is a nondegenerate vertex because

$$n = 2, \quad n_{\text{act}}(\bar{x}) = 2$$

and

$$\text{dof}(\bar{\mathbf{x}}) = 2 - 2 = 0$$

Clearly a vertex is a point where n or more independent constraints intersect in n -dimensional space to produce a point. Recall the discussion of LPs in Chapter 7; if an LP has an optimal solution, an optimal vertex (or extreme point) solution exists. Of course, this rule is not true for nonlinear problems. Optimal solutions \mathbf{x}^* of unconstrained NLPs have $\text{dof}(\bar{\mathbf{x}}) = n$, since $n_{\text{act}}(\bar{\mathbf{x}}) = 0$ (i.e., there are no constraints). Hence $\text{dof}(\bar{\mathbf{x}})$ measures how tightly constrained the point $\bar{\mathbf{x}}$ is, ranging from no active constraints ($\text{dof}(\bar{\mathbf{x}}) = n$) to completely determined by active constraints ($\text{dof}(\bar{\mathbf{x}}) \leq 0$). Degenerate vertices have “extra” constraints passing through them, that is, more than n pass through the same point. In the example, one can pass any number of redundant lines or curves through $(4, 3)$ in Figure 8.7 without affecting the feasibility of the optimal point.

If $\text{dof}(\bar{\mathbf{x}}) = n - n_{\text{act}}(\bar{\mathbf{x}}) = d > 0$, then there are more problem variables than active constraints at $\bar{\mathbf{x}}$, so the $(n - d)$ active constraints can be solved for $n - d$ dependent or basic variables, each of which depends on the remaining d independent or nonbasic variables. Generalized reduced gradient (GRG) algorithms use the active constraints at a point to solve for an equal number of dependent or basic variables in terms of the remaining independent ones, as does the simplex method for LPs.

Continuing with the example, we linearize each function about $(x_c, y_c) = (2, 2)$ and impose step bounds of 1 on both Δx and Δy , leading to the following LP:

$$\begin{aligned} \text{Maximize: } & 2x_c + y_c + 2\Delta x + \Delta y = 2\Delta x + \Delta y + 6 \\ \text{Subject to: } & x_c^2 + y_c^2 + 2x_c \Delta x + 2y_c \Delta y = 4\Delta x + 4\Delta y + 8 \leq 25 \\ & x_c^2 - y_c^2 + 2x_c \Delta x - 2y_c \Delta y = 4\Delta x - 4\Delta y \leq 7 \\ & 2 + \Delta x \geq 0, \quad 2 + \Delta y \geq 0 \\ & -1 \leq \Delta x \leq 1, \quad -1 \leq \Delta y \leq 1 \end{aligned}$$

The first two bounds require that the new point $(2 + \Delta x, 2 + \Delta y)$ satisfy the original bounds. The second two bounds, called *step bounds*, are imposed to ensure that the errors between the nonlinear problem functions and their linearizations are not too large.

Rearranging terms in the linearized LP yields the following SLP subproblem:

$$\begin{aligned} \text{Maximize: } & 2\Delta x + \Delta y \\ \text{Subject to: } & \Delta x + \Delta y \leq 4.25 \\ & \Delta x - \Delta y \leq 1.75 \end{aligned}$$

and

$$\begin{aligned} -1 & \leq \Delta x \leq 1 \\ -1 & \leq \Delta y \leq 1 \end{aligned}$$

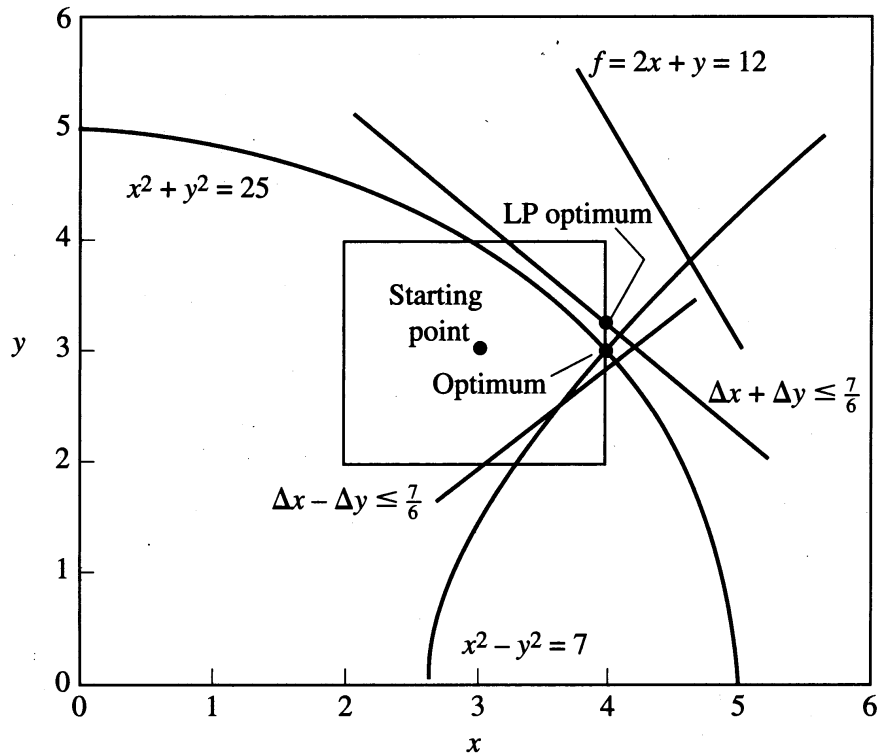


FIGURE 8.8
SLP example with linear objective, nonlinear constraints.

Figure 8.7 also shows these LP constraints. Its optimal solution is at $(\Delta x, \Delta y) = (1, 1)$, which gives $(x_n, y_n) = (3, 3)$. This point is determined entirely by the step bounds. This is an improved point, as can be seen by evaluating the original functions, so we set $x_c = x_n$ and repeat these steps to get the next LP.

$$\text{Maximize: } 2\Delta x + \Delta y$$

$$\text{Subject to: } \Delta x + \Delta y \leq \frac{7}{6}$$

$$\Delta x - \Delta y \leq \frac{7}{6}$$

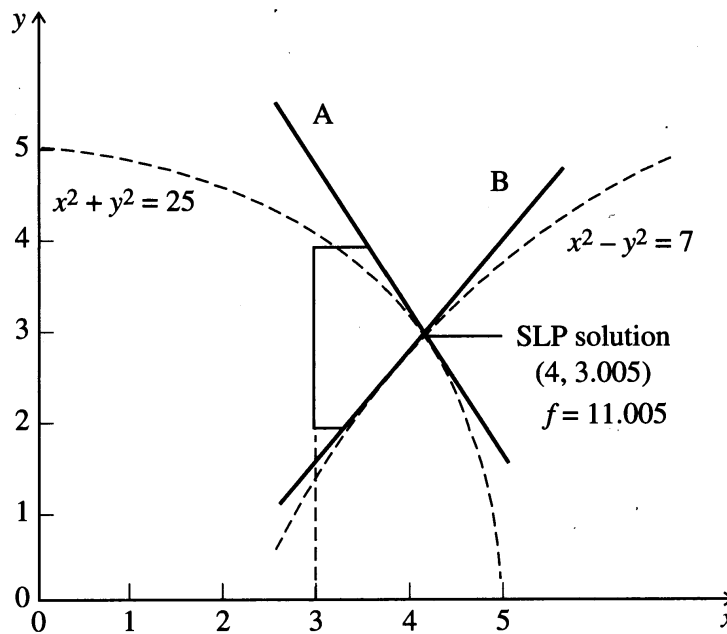
and

$$-1 \leq \Delta x \leq 1$$

$$-1 \leq \Delta y \leq 1$$

The feasible region can be seen in Figure 8.8 and the optimal solution is at $(\Delta x, \Delta y) = (1, \frac{1}{6})$ or $(x_n, y_n) = (4, 3.167)$. This point is at the intersection of the constraints $\Delta x + \Delta y \leq \frac{7}{6}$ and $\Delta x = 1$, so one step bound is still active at the LP optimum.

The SLP subproblem at $(4, 3.167)$ is shown graphically in Figure 8.9. The LP solution is now at the point $(4, 3.005)$, which is very close to the optimal point \mathbf{x}^* . This point (\mathbf{x}_n) is determined by linearization of the two active constraints, as are all further iterates. Now consider Newton's method for equation-solving applied to the two active constraints, $x^2 + y^2 = 25$ and $x^2 - y^2 = 7$. Newton's method involves

**FIGURE 8.9**

The optimal point after solving the third SLP subproblem. A is the linearization of $x^2 + y^2 = 25$ and B is the linearization of $x^2 - y^2 = 7$.

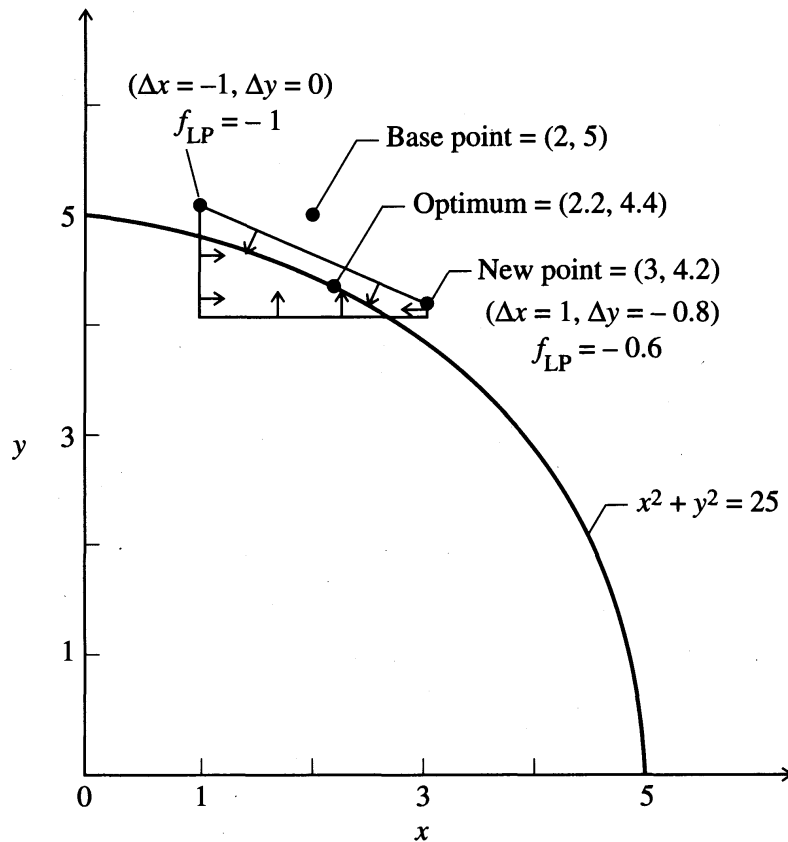
linearizing these two equations and solving for $(\Delta x, \Delta y)$, exactly as SLP is now doing. Hence, when SLP converges to a vertex optimum, it eventually becomes Newton's method applied to the active constraints. As discussed in Chapter 5, the method has quadratic convergence, that is, the new error is bounded by a constant times the previous error squared. This is the most rapid convergence we could hope to obtain, so SLP is very efficient when the optimum is at a constraint vertex.

SLP convergence is much slower, however, when the point it is converging toward is not a vertex. To illustrate, we replace the objective of the example with $x + 2y$. This rotates the objective contour counterclockwise, so when it is shifted upward, the optimum is at $\mathbf{x}^* = (2.2, 4.4)$, where only one constraint, $x^2 + y^2 \leq 25$, is active. Because the number of degrees of freedom at \mathbf{x}^* is $2 - 1 = 1$, this point is not a vertex. Figure 8.10 shows the feasible region of the SLP subproblem starting at $(2, 5)$, using step bounds of 1.0 for both Δx and Δy .

The point $(2, 5)$ is slightly infeasible, and the SLP subproblem is

$$\begin{aligned} \text{Maximize: } & f = \Delta x + 2\Delta y \\ & 4\Delta x + 10\Delta y \leq -4 \\ \text{Subject to: } & -1 \leq \Delta x \leq 1 \\ & -1 \leq \Delta y \leq 1 \end{aligned}$$

We ignore the constraint $x^2 - y^2 \leq 7$ because its linearization is redundant in this subproblem. The LP optimum is at $\Delta x = 1, \Delta y = -0.8$, so the new point is $(3, 4.2)$,

**FIGURE 8.10**

SLP subproblem at (2, 5) for the revised example ($f = x + 2y$).

which is on the “other side” of the optimum. If we continue this process without reducing the step bounds, the iterates will oscillate about the optimum and never converge to it because the new point will always be at the intersection of the linearized constraint and a step bound.

The penalty SLP algorithm (PSLP), described in Zhang et al. (1985) and discussed in the next section, contains logic for reducing the step bounds so that convergence to the optimal solution is guaranteed. The sequence of points generated by PSLP for this problem, starting at (2, 5), with initial step bounds of 0.9, is shown in Table 8.3. The algorithm converges, but much more slowly than before. The rate of convergence is linear, as occurs in the steepest descent method for unconstrained optimization. The step bounds must be reduced to force convergence, as is shown in the “max step bound” column. The significance of the “ratio” column is explained in the next section.

8.5.1 Penalty Successive Linear Programming

The PSLP algorithm is a steepest descent procedure applied to the exact L_1 penalty function (see Section 8.4). It uses a trust region strategy (see Section 6.3.2) to guar-

TABLE 8.3
Convergence of PSLP on the modified Griffith–Stewart problem

Iteration	Objective	Sum of infeasibilities	Ratio	Max step bound
0	12.000	4.000		0.900
1	11.2900	0.538	0.870	0.900
2	11.2169	0.238	0.560	0.900
3	11.2247	0.251	-0.060	0.450
4	11.1950	0.065	0.720	0.450
5	11.1821	0.064	0.020	0.225
6	11.1810	0.015	0.760	0.450
7	11.1903	0.064	-3.080	0.225
8	11.1839	0.016	-0.010	0.113
9	11.1807	3.94E-03*	0.750	0.113
10	11.1812	3.97E-03	-0.010	0.056
11	11.1805	9.86E-04	0.750	0.056
12	11.1805	9.89E-03	0.000	0.028
13	11.1804	2.46E-04	0.750	0.028
14	11.1804	2.46E-04	0.000	0.014
15	11.1803	6.16E-05	0.750	0.014
16	11.1804	2.47E-04	-3.010	0.007
17	11.1804	6.17E-05	0.000	0.003
18	11.1803	0.000	0.750	0.003
OPT	11.1803	0.000		

*E-03 represents 10^{-3} .

antee convergence. To explain PSLP, we begin with an NLP in the following general form:

$$\begin{aligned} \text{Minimize: } & f(\mathbf{x}) \\ \text{Subject to: } & \mathbf{g}(\mathbf{x}) = \mathbf{b} \end{aligned} \quad (8.51)$$

and

$$\mathbf{l} \leq \mathbf{x} \leq \mathbf{u} \quad (8.52)$$

Any inequalities have been converted to equalities using slack variables, which are included in \mathbf{x} . The exact L_1 penalty function for this problem is

$$P(\mathbf{x}, w) = f(\mathbf{x}) + w \sum_{i=1}^m |g_i(\mathbf{x}) - b_i| \quad (8.53)$$

If the penalty weight w is larger than the maximum of the absolute multiplier values for the problem, then minimizing $P(\mathbf{x}, w)$ subject to $\mathbf{l} \leq \mathbf{x} \leq \mathbf{u}$ is equivalent to minimizing f in the original problem. Often, such a threshold is known in advance, say from the solution of a closely related problem. If w is too small, PSLP will usually converge to an infeasible local minimum of P , and w can then be increased. Infeasibility in the original NLP is detected if several increases of w fail to yield a

feasible point. In the following, we drop the dependence of P on w , calling it simply $P(\mathbf{x})$.

Let \mathbf{x}^k be the value of \mathbf{x} at the start of PSLP iteration k . A piecewise linear function that closely approximates $P(\mathbf{x})$ for \mathbf{x} near \mathbf{x}^k is

$$P1(\Delta\mathbf{x}, \mathbf{x}^k) = f(\mathbf{x}^k) + \nabla f(\mathbf{x}^k)\Delta\mathbf{x} + w \sum_{i=1}^m |g_i(\mathbf{x}^k) + \nabla g_i^T(\mathbf{x}^k)\Delta\mathbf{x} - b_i| \quad (8.54)$$

As $\Delta\mathbf{x}$ approaches 0, $P1(\Delta\mathbf{x}, \mathbf{x}^k)$ approaches $P(\mathbf{x}^k)$, so $P1$ approximates P arbitrarily well if $\Delta\mathbf{x}$ is small enough. We ensure that $\Delta\mathbf{x}$ is small enough by imposing the step bounds

$$-\mathbf{s}^k \leq \Delta\mathbf{x} \leq \mathbf{s}^k \quad (8.55)$$

where \mathbf{s}^k is a vector of positive step bounds at iteration k , which are varied dynamically during the execution of PSLP. We also want the new point $\mathbf{x}^k + \Delta\mathbf{x}$ to satisfy the original bounds, so we impose the constraints

$$\mathbf{l} \leq \mathbf{x}^k + \Delta\mathbf{x} \leq \mathbf{u} \quad (8.56)$$

The trust region problem is to choose $\Delta\mathbf{x}$ to minimize $P1$ in (8.54) subject to the trust region bounds (8.55) and (8.56). As discussed in Section (8.4), this piecewise linear problem can be transformed into an LP by introducing deviation variables p_i and n_i . The absolute value terms become $(p_i + n_i)$ and their arguments are set equal to $p_i - n_i$. The equivalent LP is

Problem LP($\mathbf{x}^k, \mathbf{s}^k$)

$$\text{Minimize: } f + \nabla f^T \Delta\mathbf{x} + w \sum_i (p_i + n_i) \quad (8.57)$$

$$\text{Subject to: } g_i + \nabla g_i^T \Delta\mathbf{x} - b_i = p_i - n_i, \quad i = 1, \dots, m \quad (8.58)$$

$$-\mathbf{s}^k \leq \Delta\mathbf{x} \leq \mathbf{s}^k, \quad \mathbf{l} \leq \mathbf{x}^k + \Delta\mathbf{x} \leq \mathbf{u}, \quad p \geq 0, \quad n \geq 0, \quad i = 1, \dots, n$$

where all functions and gradients are evaluated at \mathbf{x}^k .

Let $\Delta\mathbf{x}^k$ solve LP ($\mathbf{x}^k, \mathbf{s}^k$). The new point $\mathbf{x}^k + \Delta\mathbf{x}^k$ is “better” than \mathbf{x}^k if

$$P(\mathbf{x}^k + \Delta\mathbf{x}^k) < P(\mathbf{x}^k)$$

The actual reduction in P is

$$ared_k = P(\mathbf{x}^k) - P(\mathbf{x}^k + \Delta\mathbf{x}^k)$$

Of course, $ared_k$ can be negative because P need not be reduced if the step bounds \mathbf{s}^k are too large. To decide whether \mathbf{s}^k should be increased, decreased, or left the same, we compare $ared_k$ with the reduction predicted by the piecewise linear “model” or approximation to P , $P1$. This predicted reduction is

$$pred_k = P1(0, \mathbf{x}^k) - P1(\Delta\mathbf{x}^k, \mathbf{x}^k) \quad (8.59)$$

Remember that $\Delta \mathbf{x}^k$ solves LP $(\mathbf{x}^k, \mathbf{s}^k)$, $\Delta \mathbf{x} = 0$ is feasible in this LP, and $P1$ is its objective. Because the minimal objective value is never larger than the value at any feasible solution

$$pred_k \geq 0 \quad (8.60)$$

If $pred_k = 0$, then no changes $\Delta \mathbf{x}$ within the rectangular trust region (8.58) can reduce $P1$ below the value $P1(0, \mathbf{x}^k)$. Then \mathbf{x}^k is called a stationary point of the non-smooth function P , that is, the condition $pred_k = 0$ is analogous to the condition $\nabla f(\mathbf{x}^k) = \mathbf{0}$ for smooth functions. If $pred_k = 0$, the PSLP algorithm stops. Otherwise $pred_k > 0$, so we can compute the ratio of actual to predicted reduction.

$$ratio_k = \frac{ared_k}{pred_k} \quad (8.61)$$

Changes in the step bounds are based on $ratio_k$. Its ideal value is 1.0 because then the model function $P1$ agrees perfectly with the true function P . If the ratio is close to 1.0, we increase the step bounds; if it is far from 1.0, we decrease them; and if it is in between, no changes are made. To make this precise, we set two thresholds u and l ; a ratio above u (typical value is 0.75) is “close” to 1.0, and a ratio below l (typical value is 0.25) is “far” from l . Then, the steps in PSLP iteration k are;

1. Solve the LP subproblem LP $(\mathbf{x}^k, \mathbf{s}^k)$, obtaining an optimal solution $\Delta \mathbf{x}^k$, and Lagrange multiplier estimates $\boldsymbol{\lambda}^k$. These are the LP multipliers for the equalities in (8.58).
2. Check the stopping criteria, including
 - a. $pred_k$ is nearly zero.
 - b. The KTC are nearly satisfied.
 - c. \mathbf{x}^k is nearly feasible and the fractional objective change is small enough.
3. Compute $ared_k$, $pred_k$, and $ratio_k$.
4. If $ratio_k < 0$, $\mathbf{s}^k \leftarrow \mathbf{s}^k/2$, go to step 1 (reject the new point).
5. $\mathbf{x}^k \leftarrow \mathbf{x}^k + \Delta \mathbf{x}^k$ (accept the new point).
6. If $ratio_k < l$, $\mathbf{s}^k \leftarrow \mathbf{s}^k/2$.
If $ratio_k > u$, $\mathbf{s}^k \leftarrow 2\mathbf{s}^k$.
7. Go to step 1 with $k \leftarrow k + 1$.

Step 4 rejects the new point and decreases the step bounds if $ratio_k < 0$. This step can only be repeated a finite number of times because, as the step bounds approach zero, the ratio approaches 1.0. Step 6 decreases the size of the trust region if the ratio is too small, and increases it if the ratio is close to 1.0. Zhang et al. (1986) proved that a similar SLP algorithm converges to a stationary point of P from any initial point.

Table 8.3 shows output generated by this PSLP algorithm when it is applied to the test problem of Section 8.5 using the objective $x + 2y$. This version of the problem has a nonvertex optimum with one degree of freedom. We mentioned the slow linear convergence of PSLP in this problem previously. Consider the “ratio” and “max step bound” columns of Table 8.2. Note that very small positive or negative ratios occur at every other iteration, with each such occurrence forcing a reduction

of all step bounds (they are divided by 2.0). After each reduction (once two reductions are needed), a positive ratio occurs and the new point is accepted. When the ratio is negative, the new point is rejected.

8.6 SUCCESSIVE QUADRATIC PROGRAMMING

Successive quadratic programming (SQP) methods solve a sequence of quadratic programming approximations to a nonlinear programming problem. Quadratic programs (QPs) have a quadratic objective function and linear constraints, and there exist efficient procedures for solving them; see Section 8.3. As in SLP, the linear constraints are linearizations of the actual constraints about the selected point. The objective is a quadratic approximation to the Lagrangian function, and the algorithm is simply Newton's method applied to the KTC of the problem.

Problem formulation with equality constraints

To derive SQP, we again consider a general NLP of the form (8.51)–(8.52), but temporarily ignore the bounds to simplify the explanation;

$$\text{Minimize: } f(\mathbf{x}) \quad (8.62)$$

$$\text{Subject to: } \mathbf{g}(\mathbf{x}) = \mathbf{b}$$

The Lagrangian function for this problem is

$$L(\mathbf{x}, \boldsymbol{\lambda}) = f(\mathbf{x}) + \boldsymbol{\lambda}^T(\mathbf{g}(\mathbf{x}) - \mathbf{b}) \quad (8.63)$$

and the KTC are

$$\nabla_{\mathbf{x}}L = \nabla f(\mathbf{x}) + \sum_{i=1}^m \lambda_i \nabla g_i(\mathbf{x}) = \mathbf{0} \quad (8.64)$$

and

$$\mathbf{g}(\mathbf{x}) = \mathbf{b} \quad (8.65)$$

As discussed in Section (8.2), Equations (8.64) and (8.65) is a set of $(n + m)$ nonlinear equations in the n unknowns \mathbf{x} and m unknown multipliers $\boldsymbol{\lambda}$. Assume we have some initial guess at a solution $(\bar{\mathbf{x}}, \bar{\boldsymbol{\lambda}})$. To solve Equations (8.64)–(8.65) by Newton's method, we replace each equation by its first-order Taylor series approximation about $(\bar{\mathbf{x}}, \bar{\boldsymbol{\lambda}})$. The linearization of (8.64) with respect to \mathbf{x} and $\boldsymbol{\lambda}$ (the arguments are suppressed)

$$\nabla_{\mathbf{x}}L + \nabla_{\mathbf{x}}^2L \Delta \mathbf{x} + \nabla_{\boldsymbol{\lambda}}L \Delta \boldsymbol{\lambda} = \mathbf{0} \quad (8.66)$$

and that for (8.65) is

$$\mathbf{g} + \nabla \mathbf{g} \Delta \mathbf{x} = \mathbf{0} \quad (8.67)$$

In Equations (8.66)–(8.67) all functions and derivatives are evaluated at $(\bar{\mathbf{x}}, \bar{\boldsymbol{\lambda}})$, $\nabla \mathbf{g}$ is the Jacobian matrix of \mathbf{g} , and $\nabla_x^2 L$ is the Hessian matrix of the Lagrangian.

$$\nabla_x^2 L(\bar{\mathbf{x}}, \bar{\boldsymbol{\lambda}}) = \nabla^2 f(\bar{\mathbf{x}}) + \sum_{i=1}^m \bar{\lambda}_i \nabla^2 g_i(\bar{\mathbf{x}}) \quad (8.68)$$

Note that second derivatives of all problem functions are now involved.

For problems with only equality constraints, we could simply solve the linear equations (8.66)–(8.67) for $(\Delta \mathbf{x}, \Delta \boldsymbol{\lambda})$ and iterate. To accommodate both equalities and inequalities, an alternative viewpoint is useful. Consider the quadratic programming problem

$$\text{Minimize: } \nabla L^T \Delta \mathbf{x} + \frac{1}{2} \Delta \mathbf{x}^T \nabla_x^2 L \Delta \mathbf{x} \quad (8.69)$$

$$\text{Subject to: } \mathbf{g} + \nabla \mathbf{g} \Delta \mathbf{x} = 0 \quad (8.70)$$

If we call the Lagrange multipliers for (8.70) $\Delta \boldsymbol{\lambda}$, the Lagrangian for the QP is

$$L_1(\Delta \mathbf{x}, \Delta \boldsymbol{\lambda}) = \nabla L^T \Delta \mathbf{x} + \frac{1}{2} \Delta \mathbf{x}^T \nabla_x^2 L \Delta \mathbf{x} + \Delta \boldsymbol{\lambda}^T (\mathbf{g} + \nabla \mathbf{g} \Delta \mathbf{x}) \quad (8.71)$$

Setting the derivatives of L_1 with respect to $\Delta \mathbf{x}$ and $\Delta \boldsymbol{\lambda}$ equal to zero yields the Newton equations (8.66)–(8.67) so they are the KTC of the QP (8.69)–(8.70). Hence in the equality-constrained case, we can compute the Newton step $(\Delta \mathbf{x}, \Delta \boldsymbol{\lambda})$ either by solving the linear equations (8.66)–(8.67) or by solving the QP (8.69)–(8.70).

Inclusion of both equality and inequality constraints

When the original problem has a mixture of equalities and inequalities, it can be transformed into a problem with equalities and simple bounds by adding slacks, so the problem has an objective function f , equalities (8.62), and bounds

$$\mathbf{l} \leq \mathbf{x} \leq \mathbf{u} \quad (8.72)$$

Repeating the previous development for this problem, Newton's method applied to the KTC yields a mixed system of equations and inequalities for the Newton step $(\Delta \mathbf{x}, \Delta \boldsymbol{\lambda})$. This system is the KTC for the QP in (8.69)–(8.70) with the additional bound constraints

$$\mathbf{l} \leq \bar{\mathbf{x}} + \Delta \mathbf{x} \leq \mathbf{u} \quad (8.73)$$

Hence the QP subproblem now has both equality and inequality constraints and must be solved by some iterative QP algorithm.

The approximate Hessian

Solving a QP with a positive-definite Hessian is fairly easy. Several good algorithms all converge in a finite number of iterations; see Section 8.3. However, the Hessian of the QP presented in (8.69), (8.70), and (8.73) is $\nabla_x^2 L(\bar{\mathbf{x}}, \bar{\boldsymbol{\lambda}})$, and this matrix need not be positive-definite, even if $(\bar{\mathbf{x}}, \bar{\boldsymbol{\lambda}})$ is an optimal point. In addition, to compute $\nabla_x^2 L$, one must compute second derivatives of all problem functions.

Both difficulties are eliminated by replacing $\nabla_x^2 L$ by a positive-definite quasi-Newton (QN) approximation \mathbf{B} , which is updated using only values of L and $\nabla_x L$ (See Section 6.4 for a discussion of QN updates.) Most SQP algorithms use Powell's modification (see Nash and Sofer, 1996) of the BFGS update. Hence the QP subproblem becomes

QP($\bar{\mathbf{x}}, \mathbf{B}$)

$$\text{Minimize: } \nabla L^T \Delta \mathbf{x} + \frac{1}{2} \Delta \mathbf{x}^T \mathbf{B} \Delta \mathbf{x} \quad (8.74)$$

$$\text{Subject to: } \nabla \mathbf{g} \Delta \mathbf{x} = -\mathbf{g}, \quad \mathbf{l} \leq \bar{\mathbf{x}} + \Delta \mathbf{x} \leq \mathbf{u} \quad (8.75)$$

The SQP line search

To arrive at a reliable algorithm, one more difficulty must be overcome. Newton and quasi-Newton methods may not converge if a step size of 1.0 is used at each step. Both trust region and line search versions of SQP have been developed that converge reliably [see Nocedal and Wright (1999) and Nash and Sofer (1996)]. A widely used line search strategy is to use the L_1 exact penalty function $P(\mathbf{x}, w)$ in (8.53) as the function to be minimized during the line search. This function also plays a central role in the PSLP algorithm discussed in Section 8.5. In a line search SQP algorithm, $P(\mathbf{x}, w)$ is used only to determine the step size along the direction determined by the QP solution, $\Delta \mathbf{x}$. Let \mathbf{x} be the current iterate, and let $\Delta \mathbf{x}$ solve the QP subproblem, QP(\mathbf{x}, \mathbf{B}). The L_1 exact penalty function for the NLP problem is

$$P(\mathbf{x}, w) = f(\mathbf{x}) + \sum_{i=1}^m w_i |g_i(\mathbf{x}) - b_i| \quad (8.76)$$

where a separate penalty weight w_i is used for each constraint. The SQP line search chooses a positive step size α to find an approximate minimum of

$$r(\alpha) = P(\bar{\mathbf{x}} + \alpha \Delta \mathbf{x}, w) \quad (8.77)$$

A typical line search algorithm, which uses the derivative of $r(\alpha)$ evaluated at $\alpha = 0$, denoted by $r'(0)$, is

1. $\alpha \leftarrow 1$
2. If

$$r(\alpha) < r(0) - 0.1\alpha r'(0) \quad (8.78)$$

stop and return the current α value.

3. Let α_1 be the unique minimum of the convex quadratic function that passes through $r(0)$, $r'(0)$, and $r(\alpha)$. Take the new estimate of α as

$$\alpha \leftarrow \max(0.1\alpha, \alpha_1) \quad (8.79)$$

4. Go to step 2.

This backtracking line search tries $\alpha = 1.0$ first and accepts it if the "sufficient decrease" criterion (8.78) is met. This criterion is also used in unconstrained minimization, as discussed in Section 6.3.2. If $\alpha = 1.0$ fails the test (8.78), a safe-

guarded quadratic fit (8.79) chooses the next α . The trust region in (8.79) ensures that the new α is not too small.

SQP algorithm

Based on this line search and the QP subproblem QP (\mathbf{x} , \mathbf{B}) in (8.74)–(8.75), a typical SQP algorithm follows:

1. Initialize: $\mathbf{B}^0 \leftarrow \mathbf{I}$ (or some other positive-definite matrix), $\mathbf{x}^0 \leftarrow \mathbf{x}$ (user-provided initial point), $k \leftarrow 0$.
2. Solve the QP subproblem QP (\mathbf{x}^k , \mathbf{B}^k), yielding a solution $\Delta \mathbf{x}^k$ and Lagrange multiplier estimates $\boldsymbol{\lambda}^k$.
3. Check the termination criteria (KTC, fractional objective change), and stop if any are satisfied to within the specified tolerances.
4. Update the penalty weights w in the penalty function $p(\mathbf{x}, w)$. See Nash and Sofer (1996) for details. Let the new weights be w^k .
5. Apply the line search algorithm just described to the function

$$r_k(\alpha) = P(\mathbf{x}^k + \alpha \Delta \mathbf{x}^k, w^k)$$

yielding a positive step size α^k

6. $\mathbf{x}^{k+1} = \mathbf{x}^k + \alpha^k \Delta \mathbf{x}^k$, $\boldsymbol{\lambda}^{k+1} = \boldsymbol{\lambda}^k$
7. Evaluate all problem functions and their gradients at the new point. Update the matrix \mathbf{B}^k (Nash and Sofer, 1996) using

$$L(\mathbf{x}^k), \quad L(\mathbf{x}^{k+1}), \quad \nabla_{\mathbf{x}} L(\mathbf{x}^k, \boldsymbol{\lambda}^k), \quad \nabla_{\mathbf{x}} L(\mathbf{x}^{k+1}, \boldsymbol{\lambda}^k)$$

8. Replace k by $k + 1$, and go to step 2

Convergence of SQP

Because of the quasi-Newton updating of \mathbf{B}^k , this SQP algorithm estimates second-order information, that is, \mathbf{B}^k is a positive-definite approximation of $\nabla_{\mathbf{x}}^2 L$. Hence a correctly implemented SQP algorithm can have a superlinear convergence rate, just as the BFGS algorithm for unconstrained minimization is superlinearly convergent. If the optimum is not at a vertex, SQP usually requires fewer iterations than SLP, but each iteration requires solution of a QP, which is often much slower than solving an LP (as SLP does). Hence each iteration takes longer than the corresponding SLP iteration. In addition, the approximate Hessian matrix \mathbf{B}^k is *dense*, even when the matrix it approximates, $\nabla_{\mathbf{x}}^2 L$, is sparse, so the algorithm gets slower and requires more storage (order of n^2) as the number of variables n increases. For problems with $n > 1000$, say, the SQP algorithm posed here is not practical. However, similar methods using sparse approximations to $\nabla_{\mathbf{x}}^2 L$ do exist, and these can solve much larger problems.

SQP code performance

Table 8.4 shows the convergence of an SQP algorithm very similar to the one described here, applied to the Griffith–Stewart test problem of Section 8.5, using the

TABLE 8.4
Convergence of SQP on modified
Griffith–Stewart problem

Iteration	Objective	Sum of infeasibilities
0	12.0000	4.000
1	11.2069	0.172
2	11.1810	0.015
3	11.1831	0.012
4	11.1803	2.1E-06
OPT	11.1803	0.000

objective $x + 2y$. This is the same problem as solved by PSLP in Table 8.3, using the same initial point (2, 5). Comparing the two tables shows that SQP converges much more rapidly on this problem than PSLP. This is because of the second-order information (second derivatives) estimated in the matrices \mathbf{B}^k . The price one pays for this rapid convergence is the need to store and manipulate the dense matrices \mathbf{B}^k , and to solve a more difficult subproblem (a QP instead of an LP). For problems with several thousand constraints and variables, these disadvantages usually mean that SLP is preferred. In fact, SLP is widely used in the oil and chemical industries to solve large production planning models. See Baker and Lasdon (1985) for details.

8.7 THE GENERALIZED REDUCED GRADIENT METHOD

The *generalized reduced gradient* (GRG) algorithm was first developed in the late 1960s by Jean Abadie (Abadie and Carpentier, 1969) and has since been refined by several other researchers. In this section we discuss the fundamental concepts of GRG and describe the version of GRG that is implemented in GRG2, the most widely available nonlinear optimizer [Lasdon et al., 1978; Lasdon and Waren, 1978; Smith and Lasdon, 1992].

GRG algorithms use a basic descent algorithm described below for unconstrained problems. We state the steps here:

General descent algorithm

1. Compute the gradient of $f(\mathbf{x})$ at the current point \mathbf{x}_c , giving $\nabla f(\mathbf{x}_c)$.
2. If the current point \mathbf{x}_c is close enough to being optimal, stop.
3. Compute a search direction \mathbf{d}_c using the gradient $\nabla f(\mathbf{x}_c)$ and perhaps other information such as the previous search direction.
4. Determine how far to move along the current search direction \mathbf{d}_c , starting from the current point \mathbf{x}_c . This distance α_c is most often an approximation of the value of α that minimizes the objective function $f(\mathbf{x}_c + \alpha \mathbf{d}_c)$ and is used to determine the next point $\mathbf{x}_n = (\mathbf{x}_c + \alpha_c \mathbf{d}_c)$.
5. Replace the current point \mathbf{x}_c by the next point \mathbf{x}_n , and return to step 1.

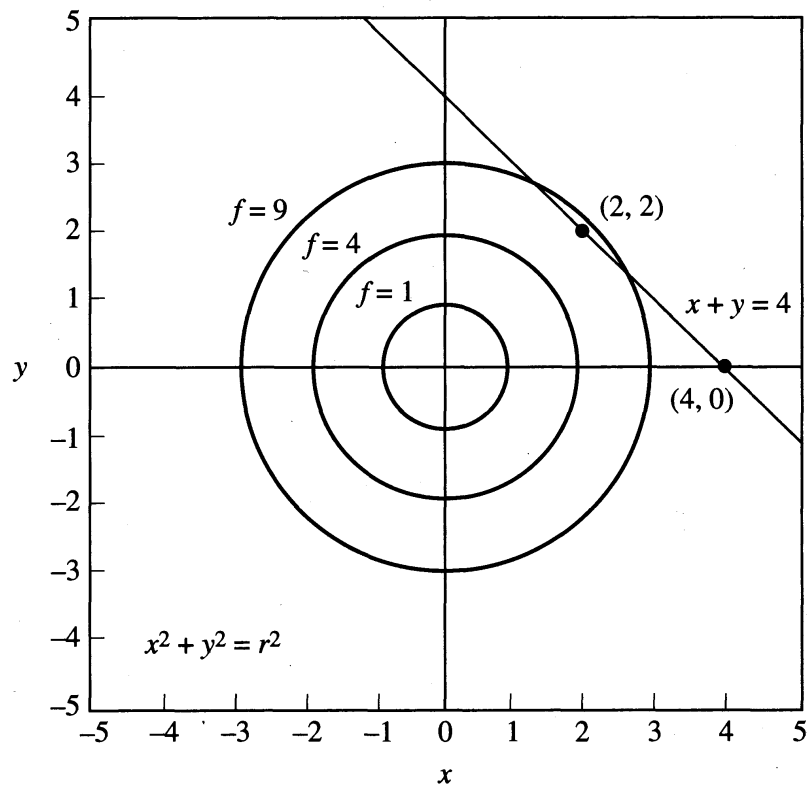


FIGURE 8.11
Circular objective contours and the linear equality constraint for the GRG example.

Equality constraints

To explain how GRG algorithms handle equality constraints, consider the following problem:

$$\text{Minimize: } x^2 + y^2$$

$$\text{Subject to: } x + y = 4$$

The geometry of this problem is shown in Figure 8.11. The linear equality constraint is a straight line, and the contours of constant objective function values are circles centered at the origin. From a geometric point of view, the problem is to find the point on the line that is closest to the origin at $x = 0, y = 0$. The solution to the problem is at $x = 2, y = 2$, where the objective function value is 8.

GRG takes a direct and natural approach to solve this problem. It uses the equality constraint to solve for one of the variables in terms of the other. For example, if we solve for x , the constraint becomes

$$x = 4 - y \quad (8.80)$$

Whenever a value is specified for y , the appropriate value for x , which keeps the equality constraint satisfied, can easily be calculated. We call y the independent, or

nonbasic, variable and x the dependent, or basic, variable. Because x is now determined by y , this problem can be reduced to one involving only y by substituting $(4 - y)$ for x in the objective function to give:

$$F(y) = (4 - y)^2 + y^2$$

The function $F(y)$ is called the reduced objective function, and the reduced problem is to minimize $F(y)$ subject to no constraints. Once the optimal value of y is found, the optimal value of x is computed from Equation (8.80).

Because the reduced problem is unconstrained and quite simple, it can be solved either analytically or by the iterative descent algorithm described earlier. First, let us solve the problem analytically. We set the gradient of $F(y)$, called the *reduced gradient*, to zero giving:

$$\begin{aligned}\nabla F(y) &= \frac{dF(y)}{dy} = -2(4 - y) + 2y \\ &= -8 + 4y = 0\end{aligned}$$

Solving this equation we get $y = 2$. Substituting this value in (8.80) gives $x = 2$ and $(x, y) = (2, 2)$ is, of course, the same solution as the geometric one.

Now apply the steps of the descent algorithm to minimize $F(y)$ in the reduced problem, starting from an initial $y_c = 0$, for which the corresponding $x_c = 4$. Computing the reduced gradient gives $\nabla F(y_c) = \nabla F(0) = -8$, which is not close enough to zero to be judged optimal so we proceed with step 3. The initial search direction is the negative reduced gradient direction, so $d = 8$ and we proceed to the line search of step 4. New points are given by

$$\begin{aligned}y &= y_c + \alpha d \\ &= 0 + 8\alpha\end{aligned}\tag{8.81}$$

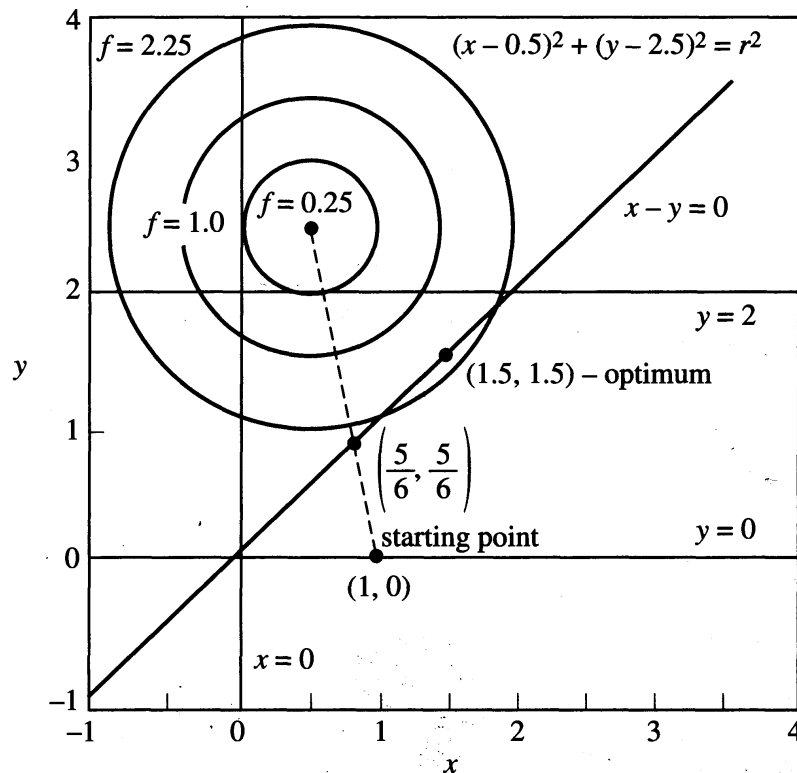
where α is the step size. We start at $(4, 0)$ with $\alpha = 0$; as α increases, y also increases. This increase is determined by Equation (8.81) and keeps (x, y) on the equality constraint shown in Figure 8.11.

Next α is selected to minimize $g(\alpha)$, the reduced objective function evaluated along the current search direction, which is given by

$$\begin{aligned}g(\alpha) &= F(y_c + \alpha d) \\ &= F(0 + 8\alpha) \\ &= (4 - 8\alpha)^2 + (8\alpha)^2\end{aligned}$$

Again, in this simple case, we can proceed analytically to determine α by setting the derivative of $g(\alpha)$ to zero to get

$$\begin{aligned}\frac{dg(\alpha)}{d\alpha} &= -16(4 - 8\alpha) + 128\alpha \\ &= -64 + 256\alpha = 0\end{aligned}$$

**FIGURE 8.12**

Circular objective function contours and linear inequality constraint.

Solving for α gives $\alpha = \frac{1}{4}$. Substituting this value into Equation (8.81) gives $y_n = 2$ and then (8.80) gives $x_n = 2$, which is the optimal solution.

Inequality constraints

Now examine how GRG proceeds when some of the constraints are inequalities and there are bounds on some or all of the variables. Consider the following problem:

$$\text{Minimize: } (x - 0.5)^2 + (y - 2.5)^2$$

$$\text{Subject to: } x - y \geq 0$$

$$0 \leq x$$

$$0 \leq y \leq 2$$

The feasible region and some contours of the objective function are shown in Figure 8.12. The goal is to find the feasible point that is closest to the point (0.5, 2.5), which is (1.5, 1.5).

GRG converts inequality constraints to equalities by introducing slack variables. If s is the slack in this case, the inequality $x - y \geq 0$ becomes $x - y - s = 0$. We must also add the bound for the slack, $s \geq 0$, giving the new problem:

$$\text{Minimize: } (x - 0.5)^2 + (y - 2.5)^2$$

$$\text{Subject to: } x - y - s = 0$$

$$0 \leq x$$

$$0 \leq y \leq 2$$

$$0 \leq s$$

Let the starting point be $(1, 0)$, at which the objective value is 6.5 and the inequality is satisfied strictly, that is, its slack is positive ($s = 1$). At this point the bounds are also all satisfied, although y is at its lower bound. Because all of the constraints (except for bounds) are inactive at the starting point, there are no equalities that must be solved for values of dependent variables. Hence we proceed to minimize the objective subject only to the bounds on the nonbasic variables x and y . There are no basic variables. The reduced problem is simply the original problem ignoring the inequality constraint. In solving this reduced problem, we do keep track of the inequality. If it becomes active or violated, then the reduced problem changes.

To solve this first reduced problem, follow the steps of the descent algorithm outlined at the start of this section with some straightforward modifications that account for the bounds on x and y . When a nonbasic variable is at a bound, we must decide whether it should be allowed to leave the bound or be forced to remain at that bound for the next iteration. Those nonbasic variables that will not be kept at their bounds are called *superbasic variables* [this term was coined by Murtaugh and Saunders (1982)]. In step 1 the reduced gradient of $f(x, y)$ is

$$\begin{aligned} \nabla F(x, y) &= \left[\frac{\partial F(x, y)}{\partial x} \quad \frac{\partial F(x, y)}{\partial y} \right]^T \\ &= [2(x - 0.5) \quad 2(y - 2.5)]^T \\ \nabla F(1, 0) &= [1 \quad -5]^T \end{aligned}$$

In this example x is a superbasic variable. To decide whether y should also be a superbasic variable and be allowed to leave its bound, examine the value of its reduced gradient component. Because this value (-5) is negative, then moving y from its bound into the feasible region, that is, increasing the value of y , decreases the objective value. We therefore consider letting y leave its bound. In GRG, a nonbasic variable at a bound is allowed to leave that bound only if (1) doing so improves the value of the objective and (2) the predicted improvement is large compared with the improvement obtained by varying only the current superbasic variables. In this example, because the magnitude of the y component of the reduced gradient is five times the magnitude of the x component, the incentive to release y from its bound is large. Thus y is added to the list of superbasic variables.

In step 3 of the descent algorithm (because the gradient is clearly not small enough to stop), the first search direction is chosen as the negative gradient direction:

$$\mathbf{d}_c = -[1 \quad -5] = [-1 \quad 5]$$

In Figure 8.12, this direction (the dashed line) points to the center of the circular objective function contours at $(0.5, 2.5)$. In step 4, the line search moves along \mathbf{d}_c

until either the objective stops decreasing or some constraint or variable bound is reached. In this example the condition that is first encountered is that the constraint $x - y \geq 0$ reaches its bound, and we then select the intersection of the search direction and the constraint $x - y = 0$ as the next point. This is the point $(\frac{5}{6}, \frac{5}{6})$ where $F = \frac{26}{9} = 2.889$.

Because we now have reached an active constraint, use it to solve for one variable in terms of the other, as in the earlier equality constrained example. Let x be the basic, or dependent, variable, and y and s the nonbasic (independent) ones. Solving the constraint for x in terms of y and the slack s yields

$$x = y + s$$

The reduced objective is obtained by substituting this relation for x in the objective function:

$$F(y, s) = (y + s - 0.5)^2 + (y - 2.5)^2$$

The reduced gradient is

$$\begin{aligned} \nabla F(y, s) &= 2[(y + s - 0.5) + (y - 2.5) \quad (y + s - 0.5)]^T \\ &= [4y + 2s - 6 \quad 2y + 2s - 1]^T \end{aligned}$$

which evaluated at $(\frac{5}{6}, 0)$ is

$$\nabla F(\frac{5}{6}, 0) = [-\frac{8}{3} \quad \frac{2}{3}]^T$$

The variable y becomes superbasic. Because s is at its lower bound of zero, consider whether s should be allowed to leave its bound, that is, be a superbasic variable. Because its reduced gradient term is $\frac{2}{3}$, increasing s (which is the only feasible change for s) increases the objective value. Because we are minimizing F , fix s at zero; this corresponds to staying on the line $x = y$. The search direction $d = \frac{8}{3}$ and new values for y are generated from

$$y = \frac{5}{6} + \frac{8}{3}\alpha$$

where α is the step size from the current point. The function to be minimized by the line search is

$$\begin{aligned} g(\alpha) &= F(y, s) = F(\frac{5}{6} + \frac{8}{3}\alpha, 0) \\ &= [\frac{1}{3} + \frac{8}{3}\alpha]^2 + [-\frac{5}{3} + \frac{8}{3}\alpha]^2 \end{aligned}$$

The optimal step size of $\alpha = \frac{1}{4}$ is determined by setting $dg(\alpha)/d\alpha = 0$, which gives the next point as $y_n = 1.5$. Because s has been fixed at zero, we are on the line $x = y$ and at step 5 we have $(x_c, y_c) = (1.5, 1.5)$, which is the optimal value for this problem. To confirm this, return to step 1 of our descent algorithm, and calculate the reduced gradient of $F(y, s)$ at $(1.5, 0)$ to get

$$\nabla F(y, s) = \nabla F(1.5, 0) = [0 \quad 1]$$

First, the first element in the reduced gradient with respect to the superbasic variable y is zero. Second, because the reduced gradient (the derivative with respect to s) is 1, increasing s (the only feasible change to s) causes an increase in the objective value. These are the two necessary conditions for optimality for this reduced problem and the algorithm terminates at $(1.5, 1.5)$ with an objective value of 2.0.

Nonlinear constraints

To illustrate how GRG handles nonlinear constraints, replace the linear constraint of the previous example by

$$(x - 2)^2 + y^2 \leq 4$$

The new problem is

$$\text{Minimize: } (x - 0.5)^2 + (y - 2.5)^2$$

$$\text{Subject to: } (x - 2)^2 + y^2 \leq 4$$

$$0 \leq x$$

$$0 \leq y \leq 2$$

The feasible region is shown in Figure 8.13. It is bounded by a semicircle of radius 2 centered at $(2, 0)$ and by the x axis. The point in this region closest to $(0.5, 2.5)$ is optimal, which is $(0.971, 1.715)$.

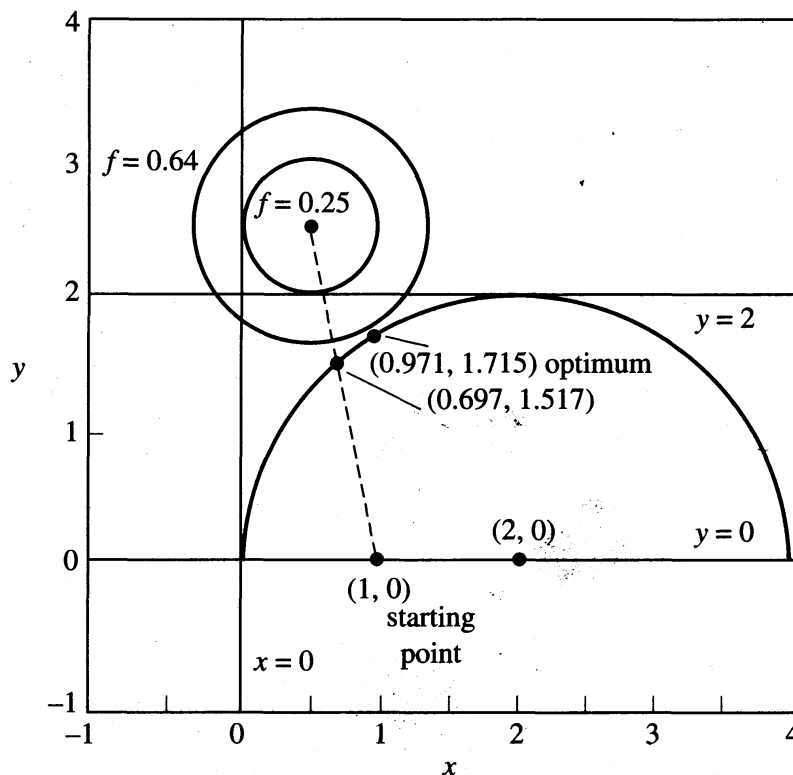


FIGURE 8.13

Circular objective function contours with a nonlinear inequality constraint.

We again start from the point (1, 0). At this point the nonlinear constraint is inactive, y is released from its lower bound to become superbasic (along with x) and progress continues along the negative gradient direction until the constraint is encountered. The intersection of the constraint and the negative gradient direction from the starting point is at (0.697, 1.715). Now the nonlinear constraint is active. Adding a slack variable s gives

$$(x - 2)^2 + y^2 + s = 4 \quad (8.82)$$

To form the reduced problem, this equation must be solved for one variable in terms of the other two. The logic in GRG for selecting which variables are to be basic is complex and is not discussed here [see Lasdon et al. (1978); and Lasdon and Waren (1978) for more information]. In this example GRG selects x as basic.

Solving (8.82) for x yields

$$x = 2 + \sqrt{4 - y^2 - s}$$

The reduced objective is obtained by substituting this expression into the objective function. The slack s will be fixed at its current zero value for the next iteration because moving into the interior of the circle from (0.697, 1.517) increases the objective. Thus, as in the linearly constrained example, y is again the only superbasic variable at this stage.

Because analytic solution of the active constraints for the basic variables is rarely possible, especially when some of the constraints are nonlinear, a numerical procedure must be used. GRG uses a variation of Newton's method which, in this example, works as follows. With $s = 0$, the equation to be solved for x is

$$(x - 2)^2 + y^2 - 4 = 0 \quad (8.83)$$

GRG determines a new value for y as before, by choosing a search direction d and then a step size α . Because this is the first iteration for the current reduced problem, the direction d is the negative reduced gradient. The line search subroutine in GRG chooses an initial value for α . At (0.697, 1.517), $d = 1.508$ and the initial value for α is 0.050. Thus the first new value for y , say y_1 , is

$$y_1 = y_c + \alpha d = 1.517 + 0.050(1.508) = 1.592$$

Substituting this value into Equation (8.83) gives

$$g(x) = (x - 2)^2 - 1.466 = 0 \quad (8.84)$$

Given an initial guess x_0 for x , Newton's method is used to solve Equation (8.84) for x by replacing the left-hand side of (8.84) by its first-order Taylor series approximation at x_0 :

$$g(x_0) + \left(\frac{\partial g(x_0)}{\partial x} \right) (x - x_0) = 0$$

Solving this equation for x and calling this result x_1 yields

$$x_1 = x_0 - \left(\frac{\partial g(x_0)}{\partial x} \right)^{-1} g(x_0) \quad (8.85)$$

If $g(x_1)$ is close enough to zero, x_1 is accepted as the solution and this procedure stops. “Close enough” is determined by a feasibility tolerance E_f (which can be set by the user, and has a default value of 0.0001) using the criterion:

$$\text{abs}[g(x_1)] \leq E_f \quad (8.86)$$

If this criterion is not satisfied, x_1 replaces x_0 , and a new iteration of Newton’s method begins. For this example, the sequence of x and y values generated by GRG is

Iteration	x	$g(x)$
Initial point	0.7849	-0.134E-01
1	0.7900	-0.940E-03
2	0.7904	-0.675E-04

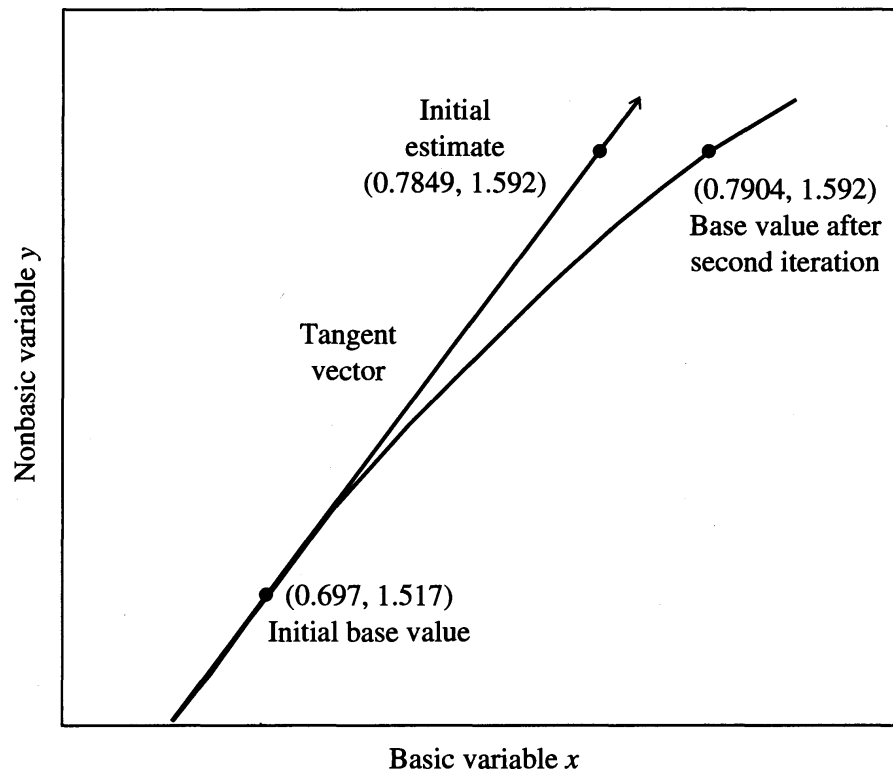
In the “pure” Newton’s method, $\partial g(x)/\partial x$ is reevaluated at each new value of x . In GRG, $\partial g(x)/\partial x$ is evaluated only once for each line search, at the point from which the line search begins. In this example, $\partial g(x)/\partial x$ evaluated at $x = 0.697$ is 2.606, so the GRG formula corresponding to (8.85) is

$$x_1 = x_0 - 0.383g(x_0)$$

This variation on Newton’s method usually requires more iterations than the pure version, but it takes much less work per iteration, especially when there are two or more basic variables. In the multivariable case the matrix $\nabla \mathbf{g}(\mathbf{x})$ (called the basis matrix, as in linear programming) replaces $\partial g/\partial \mathbf{x}$ in the Newton equation (8.85), and $\mathbf{g}(\mathbf{x}_0)$ is the vector of active constraint values at \mathbf{x}_0 .

Note that the initial guess for x in row 1 of the preceding table is 0.7849, not its base value of 0.697. GRG derives this initial estimate by using the vector that is tangent to the nonlinear constraint at (0.697, 1.517), as shown in Figure 8.14. Given $y_1 = 1.592$, the x value on this tangent vector is 0.7849. The tangent vector value is used because it usually provides a good initial guess and results in fewer Newton iterations.

Of course, Newton’s method does not always converge. GRG assumes Newton’s method has failed if more than ITLIM iterations occur before the Newton termination criterion (8.86) is met or if the norm of the error in the active constraints ever increases from its previous value (an occurrence indicating that Newton’s method is diverging). ITLIM has a default value of 10. If Newton’s method fails but an improved point has been found, the line search is terminated and a new GRG iteration begins. Otherwise the step size in the line search is reduced and GRG tries again. The output from GRG that shows the progress of the line search at iteration 4 is

**FIGURE 8.14**

Initial estimate for Newton's method to return to the nonlinear constraint.

```
STEP = 5.028E-02 OBJ = 9.073E-01 NEWTON ITERS = 2
STEP = 1.005E-01 OBJ = 8.491E-01 NEWTON ITERS = 4
STEP = 2.011E-01 OBJ = 9.128E-01 NEWTON ITERS = 8
QUADRATIC INTERPOLATION
STEP = 1.242E-01 OBJ = 8.386E-01 NEWTON ITERS = 4
```

Note that as the line search process continues and the total step from the initial point gets larger, the number of Newton iterations generally increases. This increase occurs because the linear approximation to the active constraints, at the initial point (0.697, 1.517), becomes less and less accurate as we move further from that point.

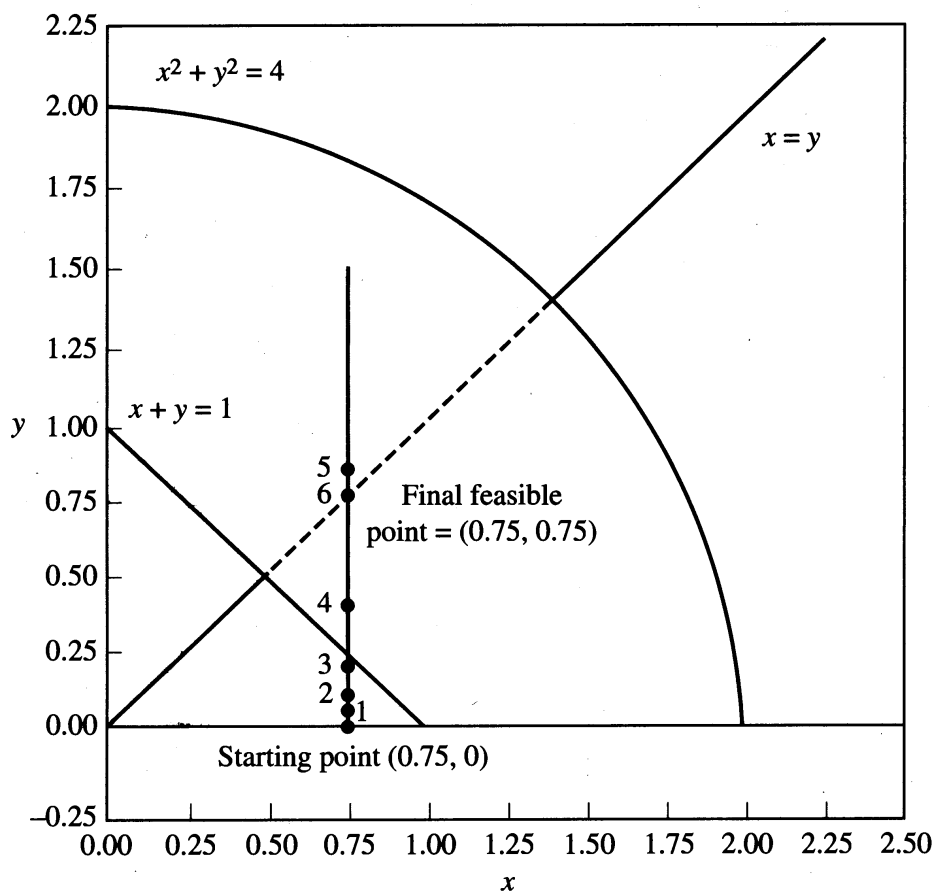
Infeasible starting point

If the initial values of the variables do not satisfy all of the constraints, GRG starts with a phase I objective function (as is also done in linear programming) and attempts to find a feasible solution. To illustrate this approach consider a problem that has no objective function and has the following three constraints:

$$x^2 + y^2 \leq 4$$

$$x + y \geq 1$$

$$x - y = 0$$

**FIGURE 8.15**

Finding a feasible point in GRG; the feasible region is the dashed line.

We use a starting point of $(0.75, 0)$. The feasible region is shown in Figure 8.15 as the dashed line segment. At the initial point constraint 1 is strictly satisfied, but constraints 2 and 3 are violated. GRG constructs the phase I objective function as the sum of the absolute values of all constraint violations. For this case the sum of the infeasibilities ($sinf$) is

$$\begin{aligned} sinf(x,y) &= (x - y) + [1 - (x + y)] \\ &= 0.75 + 0.25 \\ &= 1.0 \end{aligned}$$

The first term is the violation of constraint 2 and the second term is the violation of constraint 3. Note that both terms are arranged so that the violations are positive.

The optimization problem solved by GRG is

$$\begin{aligned} \text{Minimize: } & \quad sinf(x,y) \\ \text{Subject to: } & \quad x^2 + y^2 \leq 4 \end{aligned}$$

At the initial point, the preceding nonlinear constraint is inactive, the reduced objective is just $\text{sinf}(x, y)$, and the reduced gradient is

$$\nabla \text{sinf}(x, y) = [0 \quad -2]$$

The initial search direction is, as usual, the negative reduced gradient direction so $\mathbf{d} = [0 \quad 2]$ and we move from $(0.75, 0)$ straight up toward the line $x + y = 1$. The output from GRG is shown in the following box.

Iteration number	Objective function	Number binding	Number superbasics	Number infeasible	Norm of reduced gradient	Hessian condition
0	1.000E+00	1	2	2	2.000E+00	1.000E+00
	STEP = 2.500000E-02			OBJ = 9.00000E-01		
	STEP = 5.000000E-02			OBJ = 8.00000E-01		
	STEP = 1.000000E-01			OBJ = 6.00000E-01		
	STEP = 2.000000E-01			OBJ = 3.50000E-01		
	STEP = 4.000000E-01			OBJ = 0.00000E+00		
CONSTRAINT #3 VIOLATED BOUND						
ALL VIOLATED CONSTRAINTS SATISFIED. NOW BEGIN TO OPTIMIZE TRUE OBJECTIVE						
Iteration number	Objective function	Number binding	Number superbasics	Number infeasible	Norm of reduced gradient	Hessian condition
0	1.000E+00	1	2	0	2.000E+00	1.000E+00
KUHN-TUCKER CONDITIONS SATISFIED						

As can be seen in the output shown in the box, at the starting point (iteration 0) there are two infeasible constraints, two superbasics, and $\text{sinf} = 1$. Using the usual formula, (x, y) for the first line search is calculated as follows:

$$\begin{aligned} (x, y) &= (x_c, y_c) + \alpha(d_1, d_2) \\ &= (x_c + \alpha d_1, y_c + \alpha d_2) \\ &= (0.75 + 0\alpha, 0 + 2\alpha) \\ &= (0.75, 2\alpha) \end{aligned}$$

It is clear that the x values remain fixed at 0.75 and the y values are twice the step size at each step. In Figure 8.15 these steps are labeled 1 through 6. At step 5, GRG detects the change in sign of constraint number 3 and backs up until the constraint is binding. Because at this stage (x, y) is feasible, GRG prints the message

ALL VIOLATED CONSTRAINTS SATISFIED

If the problem had an objective function, GRG would begin minimizing the “true” objective, starting from this feasible point. Because we did not specify an objective for this problem, the algorithm stops. Minimizing *sinf* to find a feasible point, if needed, is phase I of the GRG algorithm; optimization of the true objective is phase II. If GRG cannot find a feasible solution, then phase I will terminate with a positive value of *sinf* and report that no feasible solution was found.

8.8 RELATIVE ADVANTAGES AND DISADVANTAGES OF NLP METHODS

Table 8.5 summarizes the relative merits of SLP, SQP, and GRG algorithms, focusing on their application to problems with many nonlinear equality constraints. One feature appears as both an advantage and a disadvantage—whether or not the algorithm can violate the nonlinear constraints of the problem by relatively large amounts during the solution process.

SLP and SQP usually generate points with large violations. This can cause difficulties, especially in models with log or fractional power expressions, because negative arguments for these functions may be generated. Such problems have been documented in reference to complex chemical process examples (Sarma and

TABLE 8.5
Relative merits of SLP, SQP, and GRG algorithms

Algorithm	Relative advantages	Relative disadvantages
SLP	<p>Widely used in practice</p> <p>Rapid convergence when optimum is at a vertex</p> <p>Can handle very large problems</p> <p>Does not attempt to satisfy equalities at each iteration</p> <p>Can benefit from improvements to LP solvers</p>	<p>May converge slowly on problems with nonvertex optima</p> <p>Will usually violate nonlinear constraints until convergence to optimum, often by large amounts</p>
SQP	<p>Usually requires the fewest function and gradient evaluations of all three algorithms (by far)</p> <p>Does not attempt to satisfy equalities at each iteration</p>	<p>Will usually violate nonlinear constraints until convergence, often by large amounts</p>
GRG	<p>Probably most robust of all three methods</p> <p>Versatile—especially good for unconstrained or linearly constrained problems but also works well for nonlinear constraints</p> <p>Once it reaches a feasible solution it remains feasible and then can be stopped at any stage with an improved solution</p>	<p>Needs to satisfy equalities at each step of the algorithm</p>

Reklaitis, 1982) in which SLP and some exterior penalty-type algorithms failed, but the GRG code succeeded and was quite efficient. On the other hand, algorithms that do not attempt to satisfy the equalities at each step can be faster than those that do (Berna et al., 1980). The fact that SLP and SQP satisfy any linear constraints at each iteration should ease the difficulties cited in Table 8.5 but does not eliminate them.

In some situations the optimization process must be terminated before the algorithm has reached optimality and the current point must be used or discarded. These cases usually arise in on-line process control in which time limits force timely decisions. In such cases, maintaining feasibility during the optimization process may be a requirement for the optimizer because an intermediate infeasible point makes a solution unusable.

Clearly, all three algorithms have advantages that dictate their use in certain situations. For large problems, SLP software is used most widely, because it is relatively easy to implement given a good LP code. Large-scale versions of GRG and SQP are increasingly employed, however.

8.9 AVAILABLE NLP SOFTWARE

In this section we survey implementations of the algorithms described in Sections 8.5 through 8.7. Although an increasingly large proportion of NLP users employ systems with higher level user interfaces to optimizers, such as spreadsheets and algebraic modeling systems, all such systems have at their core adaptations of one or more “stand-alone” optimization packages. By stand-alone we mean software designed specifically to accept the specification of a nonlinear program, attempt to solve it, and return the results of that attempt to the user or to an invoking application. The NLP capabilities and characteristics of those higher level systems therefore naturally derive from those of their incorporated optimizers. As a result, we begin our discussion with an overview of significant stand-alone NLP optimizers. We also illustrate, for a simple NLP problem, the inputs and outputs of some of the optimizers described later on. A comprehensive list of vendors and sources for the products discussed in this section (as well as for a large number of linear, unconstrained, and discrete optimization products) is found in Moré and Wright (1993) and Wright (2000). Advertisements for many of the systems described here can be found in the monthly magazine *OR/MS Today*, published by INFORMS (Institute for Operations Research and the Management Sciences). This magazine is an excellent source of information on analytical software of all kinds. The June 1998 issue contains an excellent NLP software survey (Nash, 1998).

8.9.1 Optimizers for Stand-Alone Operation or Embedded Applications

Most existing NLP optimizers are FORTRAN-based, although C versions are becoming more prevalent. Most are capable of operation as true stand-alone systems (the user must usually code or modify main programs and routines that return function

values) or as subsystems that are embedded in larger systems and solve problems generated by or posed through those systems. Some vendors supply source code, and others supply only object code for the customers' target platform. Details are available from the vendors as noted later on or in Moré and Wright (1993) and Wright (2000). All NLP optimizers require that the user supply the following:

- A specification of the NLP problem to be solved—at a minimum, the number of functions, the number of variables, which function is the optimization objective, bounds on the functions and variables (if different from some default scheme), and initial values of some or all variables (the system may supply default values, but using these is recommended only as a last resort).
- One or more subprograms that supply to the optimizer, on demand, the values of the functions for a specified set of variable values. Some systems also allow the user the option of supplying derivative values.

GRG-based optimizers

GRG2. This code is presently the most widely distributed for the generalized reduced gradient and its operation is explained in Section 8.7. In addition to its use as a stand-alone system, it is the optimizer employed by the “Solver” optimization options within the spreadsheet programs Microsoft Excel, Novell’s Quattro Pro, Lotus 1-2-3, and the GINO interactive solver.

In stand-alone operation, GRG2 requires the user to code a calling program in FORTRAN or C that allocates working storage and passes through its argument list the problem specifications and any nondefault values for user-modified options (an option using text files for problem specifications also exists). In addition, the user must code a subroutine that accepts as input a vector of variable values and returns a vector of function values calculated from the inputs. All constraints are assumed to be of the form

$$l_i \leq g_i(\mathbf{x}) \leq u_i$$

where l_i and u_i are (constant) lower and upper bounds.

GRG2 represents the problem Jacobian (i.e., the matrix of first partial derivatives) as a dense matrix. As a result, the effective limit on the size of problems that can be solved by GRG2 is a few hundred active constraints (excluding variable bounds). Beyond this size, the overhead associated with inversion and other linear algebra operations begins to severely degrade performance. References for descriptions of the GRG2 implementation are in Liebman et al. (1985) and Lasdon et al. (1978).

LSGRG2. This extension of GRG2 employs sparse matrix representations and manipulations and extends the practical size limit to at least 1000 variables and constraints. The interfaces to LSGRG2 are very close to those described earlier for GRG2. LSGRG2 has been interfaced to the GAMS algebraic-modeling system. Performance tests and comparisons on several large models from the GAMS library are described by Smith and Lasdon (1992).

CONOPT. This is another widely used implementation of the GRG algorithm. Like LSGRG2, it is designed to solve large, sparse problems. CONOPT is available as a stand-alone system, callable subsystem, or as one of the optimizers callable by the GAMS systems. Description of the implementation and performance of CONOPT is given by Drud (1994).

SQP-based optimizers

Implementations of the SQP algorithm described in Section 8.6 are

SQP. This is a sister code to GRG2 and available from the same source. The interfaces to SQP are very similar to those of GRG2. SQP is useful for small problems as well as large sparse ones, employing sparse matrix structures throughout. The implementation and performance of SQP are documented in Fan, et al. (1988).

NPSOL. This is a dense matrix SQP code developed at Stanford University. It is available from the same source as MINOS (see following description of MINOS). Additional details are available in Moré and Wright (1993).

NLPQL. This is another SQP implementation, callable as a subroutine and notable for its use of reverse communication. The called subsystem returns codes to the calling program, indicating what information is required on reentry. (Moré and Wright, 1993):

MINOS. This employs a modified augmented Lagrangian algorithm described in Murtagh and Saunders (1982). MINOS uses sparse matrix representations throughout and is capable of solving nonlinear problems exceeding 1000 variables and rows. MINOS is also capable of exploiting, to the greatest extent possible, the presence of purely linear variables and functions. Because the user must communicate this structure to the optimizer, the greatest utility of this feature results from coupling MINOS to higher level modeling systems that can determine problem structure. As a stand-alone system, problem specifications and user options are supplied to MINOS via an external text file, and problem Jacobian information is supplied through another file. As with the other optimizers described here, the user must supply FORTRAN routines that compute function values and, optionally, derivatives. MINOS is the default optimizer option under the GAMS system for both linear and nonlinear problems. Details for stand-alone use of MINOS and additional references are given in Murtagh and Saunders (1982).

Mathematical software libraries

Many of the major callable libraries of mathematical software include at least one general NLP component (i.e., capable of solving problems with nonlinear constraints). IMSL provides individual callable routines for most variations of

linear and nonlinear constraints and objectives. The NAG FORTRAN Library (also available as a toolbox of MATLAB) contains an SQP method for constrained problems and a variety of routines for unconstrained or specialized optimization problems. In addition, most such libraries, even those without specific constrained NLP solvers, contain routines that perform such tasks as equation solving, unconstrained optimization, and various linear algebra operations. These routines can be used as subalgorithm components to build customized NLP solvers. References for the IMSL and NAG libraries and their vendors may be found in Moré and Wright (1993).

8.9.2 Spreadsheet Optimizers

In the 1980s, a major move away from FORTRAN and C optimization began as optimizers, first LP solvers, and then NLP solvers were interfaced to spreadsheet systems for desktop computers. The spreadsheet has become, de facto, the universal user interface for entering and manipulating numeric data. Spreadsheet vendors are increasingly incorporating analytic tools accessible from the spreadsheet interface and able, through that interface, to access external databases. Examples include statistical packages, optimizers, and equation solvers.

The Excel Solver. Microsoft Excel, beginning with version 3.0 in 1991, incorporates an NLP solver that operates on the values and formulas of a spreadsheet model. Versions 4.0 and later include an LP solver and mixed-integer programming (MIP) capability for both linear and nonlinear problems. The user specifies a set of cell addresses to be independently adjusted (the decision variables), a set of formula cells whose values are to be constrained (the constraints), and a formula cell designated as the optimization objective. The solver uses the spreadsheet interpreter to evaluate the constraint and objective functions, and approximates derivatives, using finite differences. The NLP solution engine for the Excel Solver is GRG2 (see Section 8.7).

For examples that use the Excel Solver, see Chapters 7, 9, and 10. For a description of the design and use of the Excel Solver, see Fylstra, et al. (1998). An enhanced version of the Excel Solver, which can handle larger problems, is faster, and includes enhanced solvers is available from Frontline Systems—see www.frontsys.com. This website contains a wealth of information on spreadsheet optimization.

The Quattro Pro Solver. The same team that packaged and developed the Excel Solver also interfaced the same NLP engine (GRG2) to the Quattro Pro spreadsheet. Solver operation and problem specification mechanisms are similar to those for Excel.

LOTUS 123. The LOTUS 123 WINDOWS-based products incorporate linear and nonlinear solvers that operate in a fashion similar to those described earlier and use the same solver engines.

8.9.3 Algebraic Modeling Systems

An algebraic modeling system normally accepts the specification of a model in text as a system of algebraic equations. The system parses the equations and generates a representation of the expressions that can be numerically evaluated by its interpreter. In addition, some analysis is done to determine the structure of the model and to generate expressions for evaluating the Jacobian matrix. The processed model is then available for presentation to an equation solver or optimizer. The following paragraphs describe four algebraic modeling systems with NLP capabilities.

GAMS—General algebraic modeling system

The general algebraic modeling system (GAMS) allows specification and solution of large-scale optimization problems. The modeling language is algebraic with a FORTRAN-like style. The default NLP solver for GAMS is MINOS with ZOOM-XMP available for mixed-integer programming. Optional interfaces are available for most currently available LP, NLP, and MILP solvers. GAMS is available on a wide variety of platforms ranging from PCs to workstations and mainframes. Examples of GAMS models and solution output are given in Chapter 9. General references, system details, and user procedures are given in Brooke and coworkers (1992). See www.gams.com for more information.

AMPL

The main features of a mathematical programming language (AMPL) include an interactive environment for setting up and solving mathematical programs; the ability to select among several solvers; and a powerful set construct that allows for indexed, named, and nested sets. This set construct allows large-scale optimization problems to be stated tersely and in a form close to their natural algebraic expression. AMPL is described in Fourer et al. (1993). A WINDOWS version, AMPL PLUS, is available, with a graphical user interface (GUI) that greatly enhances productivity.

MPL and AIMMS

Both MPL and the advanced interactive multidimensional modeling software (AIMMS) are algebraic modeling languages operating under Microsoft Windows, with convenient GUIs; powerful modeling languages; and excellent connections to external files, spreadsheets, databases, and a wide variety of linear and nonlinear solvers. See www.maximal-usa.com for MPL, and www.aimms.com for AIMMS.

8.10 USING NLP SOFTWARE

This section addresses some of the problems with NLP optimization software. The primary determinant of solution reliability with LP solvers is numerical stability and accuracy. If the linear algebra subsystem of an LP solver is strong in these

areas, the solver will almost always terminate with one of three conditions—optimal, infeasible, unbounded—or will run up against a time or iteration limit set by the user prior to detecting one of those conditions. In contrast, many additional factors affect NLP solvers and their ability to obtain and recognize a solution.

8.10.1 Evaluation of Derivatives: Issues and Problems

All major NLP algorithms require estimation of first derivatives of the problem functions to obtain a solution and to evaluate the optimality conditions. If the values of the derivatives are computed inaccurately, the algorithm may progress very slowly, choose poor directions for movement, and terminate due to lack of progress or reaching the iteration limits at points far from the actual optimum, or, in extreme cases, actually declare optimality at nonoptimal points.

Finite difference substitutes for derivatives

When the user, whether working on stand-alone software or through a spreadsheet, supplies only the values of the problem functions at a proposed point, the NLP code computes the first partial derivatives by finite differences. Each function is evaluated at a base point and then at a perturbed point. The difference between the function values is then divided by the perturbation distance to obtain an approximation of the first derivative at the base point. If the perturbation is in the positive direction from the base point, we call the resulting approximation a forward difference approximation. For highly nonlinear functions, accuracy in the values of derivatives may be improved by using central differences; here, the base point is perturbed both forward and backward, and the derivative approximation is formed from the difference of the function values at those points. The price for this increased accuracy is that central differences require twice as many function evaluations of forward differences. If the functions are inexpensive to evaluate, the additional effort may be modest, but for large problems with complex functions, the use of central differences may dramatically increase solution times. Most NLP codes possess options that enable the user to specify the use of central differences. Some codes attempt to assess derivative accuracy as the solution progresses and switch to central differences automatically if the switch seems warranted.

A critical factor in the accuracy of finite difference approximations for derivatives is the value of the perturbation step. The default values employed by all NLP codes (generally 1.E-6 to 1.E-7 times the value of the variable) yield good accuracy when the problem functions can be evaluated to full machine precision. When problem functions cannot be evaluated to this accuracy (perhaps due to functions that are the result of iterative computations), the default step is often too small. The resulting derivative approximations then contain significant error. If the function(s) are highly nonlinear in the neighborhood of the base point, the default perturbation step may be too large to accurately approximate the tangent to the function at that point. Special care must be taken in derivative computation if the problem functions

are not closed-form functions in compiled code or a modeling language (or, equivalently, a sequence of simple computations in a spreadsheet). If each function evaluation involves convergence of a simulation, solution of simultaneous equations, or convergence of an empirical model, the interaction between the derivative perturbation step and the convergence criteria of the functions strongly affects the derivative accuracy, solution progress, and reliability. In such cases, increasing the perturbation step by two or three orders of magnitude may aid the solution process.

Analytic derivatives

Algebraic modeling systems, such as those described in Section 8.9.3, accept user-provided expressions for the objective and constraint functions and process them to produce additional expressions for the analytic first partial derivatives of these functions with respect to all decision variables. These expressions are exact, so the derivatives are evaluated to full machine precision (about 15 correct decimal digits using double precision arithmetic), and they are used by any derivative-based nonlinear code that is interfaced to the system. Finite-difference approximations to first derivatives have at most seven or eight significant digits. Hence, an NLP code used within an algebraic modeling system can be expected to produce more accurate results in fewer iterations than the same solver using finite-difference derivatives. Chemical process simulators like Aspen also compute analytic derivatives and provide these to their nonlinear optimizers. Spreadsheet solvers currently use finite-difference approximations to derivatives.

Of course, many models in chemical and other engineering disciplines are difficult to express in a modeling language, because these are usually coded in FORTRAN or C (referred to as “general purpose” programming languages), as are many existing “legacy” models, which were developed before modeling systems became widely used. General-purpose languages offer great flexibility, and models coded in these languages generally execute about ten times faster than those in an algebraic modeling system because FORTRAN and C are compiled, whereas statements in algebraic modeling systems are interpreted. This additional speed is especially important in on-line control applications (see Chapter 16).

Derivatives in FORTRAN or C models may be approximated by differencing, or expressions for the derivatives can be derived by hand and coded in subroutines used by a solver. Anyone who has tried to write expressions for first derivatives of many complex functions of many variables knows how error-prone and tedious this process is. These shortcomings motivated the development of computer programs for *automatic differentiation (AD)*. Given FORTRAN or C source code which evaluates the functions, plus the user’s specification of which variables in the program are independent, AD software augments the given program with additional statements that compute partial derivatives of all functions with respect to all independent variables. In other words, using AD along with FORTRAN or C produces a program that computes the functions and their first derivatives.

Currently, the most widely used AD codes are ADIFOR (automatic differentiation of FORTRAN) and ADIC (automatic differentiation of C). These are available

at no charge from the Mathematics and Computer Science division of Argonne National Laboratories—see www.mcs.anl.gov for information on downloading the software and further information on AD. This software has been successfully applied to several difficult problems in aeronautical and structural design as well as chemical process modeling.

8.10.2 What to Do When an NLP Algorithm Is Not “Working”

Probably the most common mode of failure of NLP algorithms is termination due to “fractional change” (i.e., when the difference in successive objective function values is a small fraction of the value itself over a set of consecutive iterations) at a point where the Kuhn–Tucker optimality conditions are far from satisfied. Sometimes this criterion is not considered, so the algorithm terminates due to an iteration limit. Termination at a significantly nonoptimal point is an indication that the algorithm is unable to make any further progress. Such lack of progress is often associated with poor derivative accuracy, which can lead to search directions that do not improve the objective function. In such cases, the user should analyze the problem functions and perhaps experiment with different derivative steps or different starting points.

Parameter adjustment

Most NLP solvers use a set of default tolerances and parameters that control the algorithm’s determination of which values are “nonzero,” when constraints are satisfied, when optimality conditions are met, and other tuning factors.

Feasibility and optimality tolerances

Most NLP solvers evaluate the first-order optimality conditions and declare optimality when a feasible solution meets these conditions to within a specified tolerance. Problems that reach what appear to be optimal solutions in a practical sense but require many additional iterations to actually declare optimality may be sped up by increasing the optimality or feasibility tolerances. See Equations (8.31a) and (8.31b) for definitions of these tolerances. Conversely, problems that terminate at points near optimality may often reach improved solutions by decreasing the optimality or feasibility tolerances if derivative accuracy is high enough.

Other “tuning” issues

The feasibility tolerance is a critical parameter for GRG algorithms because it represents the convergence tolerance for the Newton iterations (see Section 8.7 for details of the GRG algorithm). Increasing this tolerance from its default value may speed convergence of slow problems, whereas decreasing it may yield a more accurate solution (at some sacrifice of speed) or “unstick” a sequence of iterations that are going nowhere. MINOS requires specification of a parameter that penalizes constraint violations. Penalty parameter values affect the balance between seeking feasibility and improving of the objective function.

Scaling

The performance of most NLP algorithms (particularly on large problems) is greatly influenced by the relative scale of the variables, function values, and Jacobian elements. In general, NLP problems in which the absolute values of these quantities lie within a few orders of magnitude of each other (say in the range 0–100) tend to solve (if solutions exist) faster and with fewer numerical difficulties. Most codes either scale problems by default or allow the user to specify that the problem be scaled. Users can take advantage of these scaling procedures by building models that are reasonably scaled in the beginning.

Model formulation

Users can enhance the reliability of any NLP solver by considering the following simple model formulation issues:

- Avoid constructs that may result in discontinuities or undefined function arguments. Use exponential functions rather than logs. Avoid denominator terms that may tend toward zero (i.e., $1/x$ or $1/(x-1)$, etc.), multiplying out these denominators where possible.
- Be sensitive to possible “domain violations,” that is, the potential for the optimizer to move variables to values for which the functions are not defined (negative log arguments, negative square roots, negative bases for fractional exponents) or for which the functions that make up the model are not valid expressions of the systems being modeled.

Starting points

The performance of NLP solvers is strongly influenced by the point from which the solution process is started. Points such as the origin $(0, 0, \dots)$ should be avoided because there may be a number of zero derivatives at that point (as well as problems with infinite values). In general, any point where a substantial number of zero derivatives are possible is undesirable, as is any point where tiny denominator values are possible. Finally, for models of physical processes, the user should avoid starting points that do not represent realistic operating conditions. Such points may cause the solver to move toward points that are stationary points but unacceptable configurations of the physical system.

Local and global optima

As was discussed in Section 4.3, a global optimum is a feasible solution that has the best objective value. A local optimum has an objective value that is better than that of any “nearby” feasible solution. All NLP algorithms and solvers here are only capable of finding local optima. For convex programs, any local optimum is also global. Unfortunately, many NLPs are not convex or cannot be guaranteed to be convex, hence we must consider any solution returned by an NLP solver to be local. The user should examine the solution for reasonableness, perhaps re-solving the problem from several starting points to investigate what local optima exist and how these solutions differ from one another. He/she can also try a global optimizer; see Chapter 10.

REFERENCES

- Abadie, J.; and J. Carpentier. "Generalization of the Wolfe Reduced Gradient Method to the Case of Nonlinear Constraints." In *Optimization*, R. Fletcher, ed. Academic Press, New York (1969), pp. 37–47.
- Baker, T. E.; and L. Lasdon. "Successive Linear Programming at Exxon." *Manage Sci* **31**: (3), 264–274 (1985).
- Berna, T. J.; M. H. Locke; and A. W. Westerberg. "A New Approach to Optimization of Chemical Processes." *AIChE J* **26**: 37–43 (1980).
- Brooke, A.; et al. *GAMS: A User's Guide*. Boyd and Fraser, Danvers, MA (1992).
- Brown, G. G.; R. F. Dell; and R. K. Wood. "Optimization and Persistence." *Interfaces* **27** (5): 15–37 (1997).
- DiBella, C. W.; and W. F. Stevens. "Process Optimization by Nonlinear Programming." *I & E C Process Des Dev* **4**: 16–20 (1965).
- Drud, A. "CONOPT—A Large-Scale GRG-Code." *ORSA J Comput* **6** (2): Spring (1994).
- Fan, Y.; S. Sarkar; and L. Lasdon. "Experiments with Successive Quadratic Programming Algorithms." *J Optim Theory Appl* **56**: (3), 359–383 (March 1988).
- Fourer, R.; D. M. Gay; and B. W. Kernighan. *AMPL: A Modeling Language for Mathematical Programming*. Scientific Programming, San Francisco (1993).
- Fylstra, D.; L. Lasdon; J. Waton.; and A. Waven. "Design and Use of the Microsoft Excel Solver." *Interfaces* **28**: (5), 29–55 (September–October, 1998).
- Klein, M.; and R. R. Kimpel. "Application of Linearly Constrained Nonlinear Optimization to Plant to Location and Sizing." *J Indus Engin* **18**: 90 (1967).
- Lasdon, L., et al. "Design and Testing of a Generalized Relaxed Gradient Code for Nonlinear Programming." *ACM Trans Math Soft* **4**: (1) 34–50 (1978).
- Lasdon, L. S.; and A. D. Waren. "Generalized Reduced Gradient Software for Linearly and Nonlinearly Constrained Problems." *Design and Implementation of Optimization Software*, H. J. Greenberg, ed., Sijthoff and Noordhoff, Holland (1978), pp. 363–397.
- Liebman, J. F., et al. *Modeling and Optimization with GINO*. Boyd and Fraser, Danvers, MA (1986).
- Luenberger, D. G. *Linear and Nonlinear Programming*, 2nd ed. Addison-Wesley, Menlo Park, CA (1984).
- Luus, R.; and T. Jaakola. "Optimization of Nonlinear Function Subject to Equality Constraints." *Chem Process Des Develop* **12**: 380–383 (1973).
- Moré, J. J.; and S. J. Wright. *Frontiers in Applied Mathematics: Optimization Software Guide*. SIAM, Philadelphia, PA (1993).
- Murtagh, B. A.; and M. A. Saunders. "A Projected Lagrangian Algorithm and Its Implementation for Sparse Nonlinear Constraints." *Math Prog Study* **16**: 84–117 (1982).
- Nash, S. G. "Nonlinear Programming." *OR/MS Today*, 36–45 (June 1998).
- Nash, S. G.; and A. Sofer. *Linear and Nonlinear Programming*. McGraw-Hill, New York (1996).
- Nocedal, J.; and S. J. Wright. *Numerical Optimization*. Springer Series in Operations Research, New York (1999).
- Rustem, B. *Algorithms for Nonlinear Programming and Multiple Objective Functions*. Wiley, New York (1998).
- Sarma, P. V. L. N.; and G. V. Reklaitis. "Optimization of a Complex Chemical Process Using an Equation Oriented Model." *Math Prog Study* **20**: 113–160 (1982).
- Smith, S.; and L. Lasdon. "Solving Large Sparse Nonlinear Programs Using GRG." *ORSA J Comput* **4**: (1) 3–15 (Winter 1992).

- Williams, T. J.; and R. E. Otto. "A Generalized Chemical Processing Model for the Investigation of Computer Control." *A I E E Trans* **79** (Communications and Electronics) 458–473 (1960).
- Wright, S. J. "Algorithms and Software for Linear and Nonlinear Programming. FOCAPD Proceedings, *AIChE Symp Ser* **96**: 323, 58–69 (2000).
- Zhang, J.; N. Kim; and L. Lasdon. "An Improved Successive Linear Programming Algorithm." *Manage Sci* **31**: 1312–1331 (1985).

SUPPLEMENTARY REFERENCES

- Bhatia, T. K.; and L. T. Biegler. "Multiperiod Design and Planning with Interior Point Methods." *Comp Chem Engin* **23**: 919 (1999).
- de Gouvêa, M. T.; and D. Odloak. "A New Treatment of Inconsistent Quadratic Programs in an SQP-Based Algorithm." *Comp Chem Engin* **22**: 1623 (1998).
- Dennis, J. E. Jr.; and R. B. Schnabel. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. SIAM, Philadelphia, PA (1996).
- Duvall, P. M.; and J. B. Riggs. "On-line Optimization of the Tennessee Eastman Challenge Problem." *J Proc Control* **10**: 19 (1999).
- Rustem, B. *Algorithms for Nonlinear Programming and Multiple Objective Decisions*. Wiley, New York (1998).
- Ternet, D. J.; and L. T. Biegler. "Recent Improvements to a Multiplier-free Reduced Hessian Successive Quadratic Programming Algorithm." *Comp Chem Engin* **22**: 963 (1998).
- Turton, R.; R. C. Bailie; W. B. Whiting; and J. Shaewitz. *Analysis, Synthesis, and Design of Chemical Processes*. Prentice-Hall, Upper Saddle River, NJ (1998).
- Vassiliadis, V. S.; and S. A. Brooks. "Application of the Modified Barrier Method in Large-Scale Quadratic Programming Problems." *Comp Chem Engin* **22**: 1197–1205 (1998).

PROBLEMS

8.1 Solve

$$\text{Minimize: } -x_1$$

$$\text{Subject to: } x_1 + x_2^4 = 0$$

by solving the constraint for x_1 and substituting into the objective function. Do you get $\mathbf{x}^* = [0 \ 0]^T$?

8.2 Solve

$$\text{Minimize: } -x_1^2$$

$$\text{Subject to: } 10^{-5}x_2^2 + x_1 = 1$$

by solving the constraint for x_1 and substituting into the objective function. Do you get $\mathbf{x}^* = [1 \ 0]^T$?

- 8.3** Explain in *no more* than three sentences how the nonlinear inequality constraints in a nonlinear programming problem can be converted into equality constraints. Demonstrate for $g(\mathbf{x}) = x_1x_2 + x_2^2 + e^{x_3} \leq 4$.
- 8.4** Use the method of Lagrange multipliers to solve the following problem. Find the values of x_1 , x_2 , and ω that

$$\begin{aligned} \text{Minimize: } & f(\mathbf{x}) = x_1^2 + x_2^2 \\ \text{Subject to: } & h(\mathbf{x}) = 2x_1 + x_2 - 2 = 0 \end{aligned}$$

- 8.5** Solve the following problem via the Lagrange multiplier method:
Find the maximum and minimum distances from the origin to the curve

$$5x_1^2 + 6x_1x_2 + 5x_2^2 = 8$$

Hint: The distance $\sqrt{x_1^2 + x_2^2}$ is the objective function.

- 8.6** Show that Lagrange multipliers do not exist for the following problem:

$$\begin{aligned} \text{Minimize: } & f(\mathbf{x}) = x_1^2 + x_2^2 \\ \text{Subject to: } & (x_1 - 1)^3 - x_2^2 = 0 \end{aligned}$$

- 8.7** Examine the reactor in Figure P8.7. The objective function, $f(c, T) = (c - c_r)^2 + T^2$ is subject to the constraint $c = c_0 + e^T$ and also $c_0 < K$, where c_r is the set point for the outlet concentration, a constant, and K is a constant.

Find the minimum value of the objective function using Lagrange multipliers for the case in which $K = c_r - 2$.

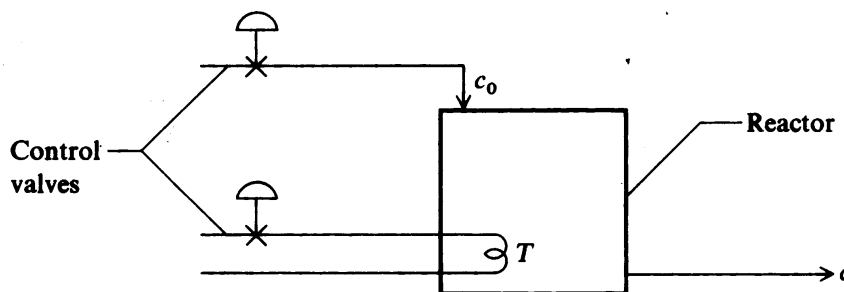


FIGURE P8.7

- 8.8** Examine the continuous through-circulation dryer problem posed by Luus and Jaakola (1973):

$$\begin{aligned} \text{Maximize: } & P \text{ given by} \\ & P = 0.0064x_1[1 - \exp(-0.184x_1^{0.3}x_2)] \\ \text{Subject to: } & \text{the power constraint} \\ & (3000 + x_1)x_1^2x_2 = 1.2 \times 10^{13} \end{aligned}$$

and the moisture content distribution constraint

$$\exp(0.184x_1^{0.3}x_2) = 4.1$$

They obtained the solution $\mathbf{x}^* = [31,766 \ 0.342]$ $P = 153.71$. Does this problem satisfy the first-order conditions?

Repeat for the problem of minimizing the capital investment for batch processes. The problem is to choose x_1 , x_2 , and x_3 to minimize

$$\begin{aligned} P = & 592V^{0.65} + 582V^{0.39} + 1200V^{0.52} + 370\left(\frac{V}{x_1}\right)^{0.22} \\ & + 250\left(\frac{V}{x_2}\right)^{0.40} + 210\left(\frac{V}{x_2}\right)^{0.62} + 250\left(\frac{V}{x_3}\right)^{0.40} \\ & + 200\left(\frac{V}{x_3}\right)^{0.85} \end{aligned}$$

subject to the simple constraint

$$V = 50(10 + x_1 + x_2 + x_3)$$

They obtained the solution $P = 126,302.9$ and

$$\mathbf{x}^* = [0.11114 \ 1.46175 \ 3.42476]$$

8.9 Maximize: $f = x_1^2 + x_2^2 + 4x_1x_2$

Subject to: $x_1 + x_2 = 8$

- Form the Lagrangian L . Set up the necessary conditions for a maximum, and solve for the optimum.
- If the constraint is changed to $x_1 + x_2 = 8.01$, compute f and L without resolving as in part a.

8.10 (a) Minimize $f = x_1^2 + x_2^2 + 10x_1 + 20x_2 + 25$

Subject to: $x_1 + x_2 = 0$

using the Lagrange multiplier technique. Calculate the optimum values of x_1, x_2, λ , and f .

- Using sensitivity analysis, determine the increase in f^{opt} when the constraint is changed to $x_1 + x_2 = 0.01$.
- Let the constraint be added to f by a penalty function:

$$P = f + r(x_1 + x_2)^2$$

Find the optimum of P with respect to x_1 and x_2 (an unconstrained problem), noting that x_1^* and x_2^* are functions of r .

- Is there a relationship between r, x_1^*, x_2^* , and λ^* ?
- Perform the second derivative test on P ; is it convex for $P \gg 1$?

8.11 Is the problem

$$\text{Minimize: } f(\mathbf{x}) = x_1^3 + 4x_2^2 - 4x_1$$

$$\text{Subject to: } 2x_2 - x_1 \geq 12$$

a convex programming problem?

8.12 Determine whether the vector $x^T = [0 \ 0]$ is an optimal solution of the problem

$$\text{Minimize: } f(\mathbf{x}) = (x_1 - 1)^2 + x_2^2$$

$$\text{Subject to: } h(\mathbf{x}) = x_1^2 + x_2^2 + x_1 + x_2 = 0$$

$$g(\mathbf{x}) = -x_1 + x_2^2 \geq 0$$

8.13 Determine whether the point $x = [0 \ 0 \ 0]^T$ is a local minimum of the problem:

$$\text{Minimize: } f(\mathbf{x}) = \frac{4}{3}(x_1^2 - x_1x_2)^{3/4} + x_3$$

$$\text{Subject to: } x_1^2 + x_2^2 + x_3^2 = 0$$

$$x_1 \geq 0$$

$$x_2 \geq 0$$

$$x_3 \geq 0$$

Show all computations.

8.14 Test whether the solution $\mathbf{x}^* = [2 \ 2]^T$ meets the sufficient conditions for a minimum of the following problem.

$$\text{Minimize: } f(\mathbf{x}) = -x_1^2x_2$$

$$\text{Subject to: } h_1(\mathbf{x}) = x_1x_2 + \left(\frac{x_1^2}{2}\right) = 6$$

$$g_2(\mathbf{x}) = x_1 - x_2 \geq 0$$

8.15 Do (a) the necessary and (b) the sufficient conditions hold at the optimum $\mathbf{x}^* = [0.82 \ 0.91]^T$ for the following problem?

$$\text{Minimize: } f(\mathbf{x}) = (x_1 - 2)^2 + (x_2 - 1)^2$$

$$\text{Subject to: } g_1(\mathbf{x}) = -\frac{x_1^2}{4} - x_2^2 + 1 \geq 0$$

$$h_2(\mathbf{x}) = x_1 - 2x_2 + 1 = 0$$

Note: $\mathbf{x}^* = \left[\left(-1 + \sqrt{7} \right) / 2 \quad \left(1 + \sqrt{7} \right) / 4 \right]^T$ exactly.

8.16 Does the following solution $\mathbf{x}^* = \left[\frac{1}{3} \quad \frac{5}{3}\right]^T$ meet the sufficient conditions for a minimum of the following problem?

$$\text{Minimize: } f(\mathbf{x}) = -\ln(1 + x_1) - \ln(1 + x_2)^2$$

$$\text{Subject to: } g_1(\mathbf{x}) = x_1 + x_2 - 2 \leq 0$$

$$g_2(\mathbf{x}) = x_1 \geq 0$$

$$g_3(\mathbf{x}) = x_2 \geq 0$$

8.17 Solve the following problems via a quadratic programming code.

P8.4 P8.9 P8.20 P8.22a

8.18 Find the stationary point of the function $f(\mathbf{x}) = x_1^2 + x_2^2 + 4x_1x_2$ subject to the constraint $x_1 + x_2 = 8$. Use direct substitution. What kind of stationary point is it?

For the same objective function and constraint, form a new function

$$P = x_1^2 + x_2^2 + 4x_1x_2 + r(x_1 + x_2 - 8)^2$$

where r is a large number. Then optimize P .

- Find the stationary point of P with respect to x_1 and x_2 , solving for x_1^* and x_2^* in terms of r .
- Find x_1^*, x_2^* as $r \rightarrow \infty$
- Does $P^* \rightarrow f^*$ for $r \rightarrow \infty$?

8.19 Minimize: $x_2^2 - x_1^2$

$$\text{Subject to: } x_1^2 + x_2^2 = 4.$$

- Use Lagrange multipliers
- Use a penalty function.

8.20 The problem is to

$$\text{Minimize: } f(\mathbf{x}) = x_1^2 + 6x_1 + x_2^2 + 9$$

$$\text{Subject to: } g_i(\mathbf{x}) = x_i \geq 0, \quad \text{for } i = 1, 2$$

From the starting vector $\mathbf{x}^0 = [1 \quad 0.5]^T$.

- Formulate a penalty function suitable to use for an unconstrained optimization algorithm.
- Is the penalty function convex?

8.21 A statement in a textbook is

The penalty term of an augmented Lagrangian method is designed to add positive curvature so that the Hessian of the augmented function is positive-definite.

Is this statement correct?

8.22 Formulate the following problems as

- (a) Penalty function problems
 (b) Augmented Lagrangian problems

(1) Minimize: $f(\mathbf{x}) = 2x_1^2 - 2x_1x_2 + 2x_2^2 - 6x_1 + 6$
 Subject to: $h(\mathbf{x}) = x_1 + x_2 - 2 = 0$

(2) Minimize: $f(\mathbf{x}) = x_1^3 - 3x_1x_2 + 4$
 Subject to: $g(\mathbf{x}) = 5x_1 + 2x_2 \geq 18$
 $h(\mathbf{x}) = -2x_1 + x_2^2 - 5 = 0$

8.23 Comment on the following proposed penalty functions suggested for use with the problem.

Minimize: $f(\mathbf{x})$
 Subject to: $g_i(\mathbf{x}) \geq 0, \quad i = 1, 2, \dots, m$

starting from a feasible point. The P functions are

(a)

$$P(\mathbf{x}, \mathbf{x}^k) = \frac{1}{f(\mathbf{x}^k) - f(\mathbf{x})} + \sum_{j=1}^m \frac{1}{g_j(\mathbf{x})}$$

(b)

$$P(\mathbf{x}, r) = f(\mathbf{x}) - r \sum_{j=1}^m \ln g_j(\mathbf{x})$$

(c)

$$P(\mathbf{x}, r^k) = f(\mathbf{x}) + r^k \sum_{j=1}^m \frac{1}{g_j(\mathbf{x})}$$

What advantages might they have compared with one another? What disadvantages?

8.24 The problem of optimizing production from several plants with different cost structures and distributing the products to several distribution centers is common in the chemical industry. Newer plants often yield lower cost products because we learn from the mistakes made in designing the original plant. Due to plant expansions, rather unusual cost curves can result. The key cost factor is the incremental variable cost, which gives the cost per pound of an additional pound of product. Ordinarily, this variable cost is a function of production level.

Consider three different plants producing a product called DAB. The Frag plant located in Europe has an original design capacity of 100×10^6 lb/year but has been expanded to produce as high as 170×10^6 lb/year. The incremental variable cost for this plant decreases slightly up to 120×10^6 lb/year, but for higher production rates severe reaction conditions cause the yields to deteriorate, causing a gradual increase in the variable cost, as shown by the following equation. No significant byproducts are

sold from this plant. Using $VC =$ variable cost is $\$/100$ lb and $x =$ production level $\times 10^{-6}$ lb/year

$$VC = 4.5 - (x - 100)(0.005), \quad 100 \leq x \leq 120$$

$$VC = 4.4 + (x - 120)(0.02), \quad 120 \leq x \leq 170$$

The Swung-Lo plant, located in the Far East, is a relatively new plant with an improved reactor/recycle design. This plant can be operated between 80×10^6 and 120×10^6 lb/year and has a constant variable cost of $\$/5.00/100$ lb.

The Hogshooter plant, located in the United States, has a range of operation from 120×10^6 to 200×10^6 lb/year. The variable cost structure is rather complicated due to the effects of extreme reaction conditions, separation tower limitation, and several byproducts, which are affected by environmental considerations. These considerations cause a discontinuity in the incremental variable cost curve at 140×10^6 lb/year as given by the following equations:

$$VC = 3.9 + (x - 120)(0.005), \quad 120 \leq x \leq 140$$

$$VC = 4.6 + (x - 140)(0.01), \quad 140 \leq x \leq 200$$

The three main customers for the DAB are located in the Europe (C1), the Far East (C2), and the United States (C3), respectively. The following matrix shows the transportation costs of ($\$/lb$) and total demand to the customers (C1, C2, C3) with plant locations denoted as A1 (Frag), A2 (Swung-Lo), and A3 (Hogshooter). The closest pairing geographically is A1-C1; A2-C2; and A3-C3.

	A1	A2	A3	Total demand
C1	0.2	0.7	0.6	140
C2	0.7	0.3	0.8	100
C3	0.6	0.8	0.2	170

Use an iterative method based on successive linearization of the objective function to determine the optimum distribution plan for the product, DAB. Use an LP code to minimize total cost at each iteration.

8.25 Maximize: $f(\mathbf{x}) = 0.5(x_1x_4 - x_2x_3 + x_3x_9 - x_5x_9 + x_5x_8 - x_6x_7)$

Subject to: $1 - x_3^2 - x_4^2 \geq 0$

$$1 - x_9^2 \geq 0$$

$$1 - x_5^2 - x_6^2 \geq 0$$

$$1 - x_1^2 - (x_2 - x_9)^2 \geq 0$$

$$1 - (x_1 - x_5)^2 - (x_2 - x_6)^2 \geq 0$$

$$1 - (x_1 - x_7)^2 - (x_2 - x_8)^2 \geq 0$$

$$1 - (x_3 - x_5)^2 - (x_4 - x_6)^2 \geq 0$$

$$1 - (x_3 - x_7)^2 - (x_4 - x_8)^2 \geq 0$$

$$1 - x_7^2 - (x_8 - x_9)^2 \geq 0$$

$$x_1x_4 - x_2x_3 \geq 0$$

$$x_3x_9 \geq 0$$

$$-x_5x_9 \geq 0$$

$$x_5x_8 - x_6x_7 \geq 0$$

$$x_9 \geq 0$$

Starting point: $x_0^i = 1, \quad i = 1, 9$

Solve using an SQP code.

8.26 Solve the following over-constrained problem.

$$\text{Minimize:} \quad f(\mathbf{x}) = x_1^2 + x_2^2 + x_3^2$$

$$\text{Subject to:} \quad g_1(\mathbf{x}) = -2x_1 - x_2 \geq -5$$

$$g_2(\mathbf{x}) = -x_1 - x_3 \geq -2$$

$$g_3(\mathbf{x}) = -x_1 - 2x_2 - x_3 \geq -10$$

$$h_1(\mathbf{x}) = 2x_1 - 2x_2 + x_3 = -2$$

$$h_2(\mathbf{x}) = 10x_1 + 8x_2 - 14x_3 = 26$$

$$h_3(\mathbf{x}) = -4x_1 + 5x_2 - 6x_3 = 6$$

$$x_1 \geq 1 \quad x_2 \geq 2 \quad x_3 \geq 0$$

$$\text{Starting point:} \quad \mathbf{x}^0 = [1 \quad 1 \quad 1]^T$$

Use successive quadratic programming.

8.27 Solve the following problems by the generalized reduced-gradient method. Also, count the number of function evaluations, gradient evaluations, constraint evaluations, and evaluations of the gradient of the constraints.

$$(a) \quad \text{Minimize:} \quad f(\mathbf{x}) = -(x_1^2 + x_2^2 + x_3^2)$$

$$\text{Subject to:} \quad x_1 + 2x_2 + 3x_3 - 1 = 0$$

$$x_1^2 + \frac{x_2^2}{2} + \frac{x_3^2}{4} - 4 = 0$$

Use various starting points.

$$\mathbf{x}^0 = [n \ n \ n], \quad \text{where } n = 2, 4, 6, 8, 10, -2, -4, -6, -8, -10$$

$$(b) \quad \text{Minimize:} \quad f(\mathbf{x}) = (x_1 - 1)^2 + (x_1 - x_2)^2 + (x_2 - x_3)^2 \\ + (x_3 - x_4)^4 + (x_4 - x_5)^4$$

$$\text{Subject to: } x_1 + x_2^2 + x_3^3 - 2 - 3\sqrt{2} = 0$$

$$x_2 - x_3^2 + x_4 + 2 - 2\sqrt{2} = 0$$

$$(x_1)(x_5) - 2 = 0$$

$$\mathbf{x}^0 = [n \ n \ n \ n \ n]^T, \text{ where } n = 2, 4, 6, 8, 10, -2, -4, -6, -8, -10$$

8.28 At stage $k = 2$, the generalized reduced-gradient method is to be applied to the following problem at the point $\mathbf{x} = [0 \ 1 \ 1]^T$.

$$\text{Minimize: } f(\mathbf{x}) = 2x_1^2 + 2x_2^2 + x_3^2 - 2x_1x_2 - 4x_1 - 6x_2$$

$$\text{Subject to: } x_1 + x_2 + x_3 = 2$$

$$x_1^2 + 5x_2 = 5$$

$$x_i \geq 0, \quad i = 1, 2, 3$$

- Compute the component step direction (+ or -) and value of each of the three variables after searching in the selected direction.
- Reduce $f(\mathbf{x})$ in the search direction.

Explain (only) in detail how you would reach the feasible point to start the next stage ($k = 3$) of optimization.

8.29 Answer true or false:

- In the generalized reduced-gradient method of solving NLP problems, the nonlinear constraints and the objective function are repeatedly linearized.
- Successive quadratic programming is based on the application of Newton's method to some of the optimality conditions for the Lagrangian function of the problem, that is, the sum of the objective function and the product of the Lagrangian multipliers times the equality constraints.

8.30 Solve the following problems using

- A generalized reduced-gradient code
- A successive quadratic programming code. Compare your results.

$$(a) \text{ Minimize: } f(\mathbf{x}) = \sum_{k=1}^{10} \left(\frac{1}{k} x_k^2 + kx_k + k^2 \right)^2$$

$$\text{Subject to: } h_1(\mathbf{x}) = x_1 + x_3 + x_5 + x_7 + x_9 = 0$$

$$h_2(\mathbf{x}) = x_2 + 2x_4 + 3x_6 + 4x_8 + 5x_{10} = 0$$

$$h_3(\mathbf{x}) = 2x_2 - 5x_5 + 8x_8 = 0$$

$$g_1(\mathbf{x}) = -x_1 + 3x_4 - 5x_7 + x_{10} \geq 0$$

$$g_2(\mathbf{x}) = -x_1 - 2x_2 - 4x_4 - 8x_8 \geq -100$$

$$g_3(\mathbf{x}) = -x_1 - 3x_3 - 6x_6 + 9x_9 \geq -50$$

$$-10^3 \leq x_i \leq 10^3, \quad i = 1, 2, \dots, 10$$

Starting point (feasible): $x_i^0 = 0, \quad i = 1, 2, \dots, 10$

$$f(\mathbf{x}^0) = 25,333.0$$

(b) Minimize: $f(\mathbf{x}) = \sum_{i=1}^{11} x_i + \sum_{i=1}^{10} (x_i + x_{i+1})$

Subject to: $x_i \geq 0, \quad i = 1, \dots, 11$

$$h_1(\mathbf{x}) = 0.1x_1 + 0.2x_7 + 0.3x_8 + 0.2x_9 + 0.2x_{11} = 1.0$$

$$h_2(\mathbf{x}) = 0.1x_2 + 0.2x_8 + 0.3x_9 + 0.4x_{10} + 1.0x_{11} = 2.0$$

$$h_3(\mathbf{x}) = 0.1x_3 + 0.2x_8 + 0.3x_9 + 0.4x_{10} + 2.0x_{11} = 3.0$$

$$g_4(\mathbf{x}) = x_4 + x_8 + 0.5x_9 + 0.5x_{10} + 1.0x_{11} \geq 1.0$$

$$g_5(\mathbf{x}) = 2.0x_5 + x_6 + 0.5x_7 + 0.5x_8 + 0.25x_9 + 0.25x_{10} + 0.5x_{11} \geq 1.0$$

$$g_6(\mathbf{x}) = x_4 + x_6 + x_8 + x_9 + x_{10} + x_{11} \geq 1.0$$

$$g_7(\mathbf{x}) = 0.1x_1 + 1.2x_7 + 1.2x_8 + 1.4x_9 + 1.1x_{10} + 2.0x_{11} \geq 1.0$$

Starting point (feasible): $x_i = 1.0, \quad i = 1, 2, \dots, 11$

(c) Maximize: $f(\mathbf{x}) = 3x_1 e^{-0.1x_1 x_6} + 4x_2 + x_3^2 + 7x_4 + \frac{10}{x_5} + x_6$

Subject to: $-x_4 + x_5 - x_6 = 0.1$

$$x_1 + x_2 + x_3 + x_4 + x_5 + x_6 = 10$$

$$2x_1 + x_2 + x_3 + 3x_4 \geq 2$$

$$-8x_1 - 3x_2 - 4x_3 + x_4 - x_5 \geq -10$$

$$-2x_1 - 6x_2 - x_3 - 3x_4 - x_6 \geq -13$$

$$-x_1 - 4x_2 - 5x_3 - 2x_4 \geq -18$$

$$-20 \leq x_i \leq 20, \quad i = 1, \dots, 6$$

Starting point (nonfeasible): $x_i = 1.0, \quad i = 1, \dots, 6$

(d) Minimize: $f(\mathbf{x}) = x_1^2 + 2x_2^2 + 3x_3^2 + 4x_4^2 + 5x_5^2$

Subject to: $h_1(\mathbf{x}) = 2x_1 + x_2 - 4x_3 + x_4 - x_5 = 0$

$$h_2(\mathbf{x}) = 5x_1 - 2x_3 + x_4 - x_5 = 0$$

$$g_1(\mathbf{x}) = x_1 + 2x_2 + x_3 \geq 6$$

$$g_2(\mathbf{x}) = 4x_3 + x_4 - 2x_5 \leq 0$$

Starting point (nonfeasible): $x_i = 1, i = 1, \dots, 5$

(e) Minimize: $f(\mathbf{x}) = (x_1 - x_2)^2 + (x_2 - x_3)^2 + (x_3 - x_4)^4 + (x_4 - x_5)^4$

Subject to: $x_1 + 2x_2 + 3x_3 - 6 = 0$

$$x_2 + 2x_3 + 3x_4 - 6 = 0$$

$$x_3 + 2x_4 + 3x_5 - 6 = 0$$

Starting point (feasible): $\mathbf{x}^0 = [35 \quad -31 \quad 11 \quad 5 \quad -5]^T$

(f) Minimize: $f(\mathbf{x}) = (x_1 - 1)^2 + (x_1 - x_2)^2 + (x_3 - 1)^2 + (x_3 - 1)^2$
 $+ (x_4 - 1)^4 + (x_5 - 1)^6$

Subject to: $x_1^2 x_4 + \sin(x_4 - x_5) - 2\sqrt{2} = 0$

$$x_2 + x_3^4 x_4^2 - 8 - \sqrt{2} = 0$$

Starting point: $\mathbf{x}^0 = [2 \quad 2 \quad 2 \quad 2 \quad 2]^T$

(g) Minimize: $f(\mathbf{x}) = (x_1 - 1)^2 + (x_1 - x_2)^2 + (x_2 - x_3)^2 + (x_3 - x_4)^4 + (x_4 - x_5)^4$

Subject to: $x_1 + x_2^2 + x_3^3 - 2 - 3\sqrt{2} = 0$

$$x_2 - x_3^2 + x_4 + 2 - 2\sqrt{2} = 0$$

$$x_1 x_5 - 2 = 0$$

Starting points: $x_1 = \pm 10, x_2 = \pm 8, x_3 = \pm 6, x_4 = \pm 4, x_5 = \pm 2$

8.31 Explain in *no more* than three sentences how an initially feasible starting point can be obtained in solving a nonlinear programming problem. Demonstrate on the problem

Maximize: $f(\mathbf{x}) = \left[(1 + x_1)^2 + x_2^2 \right]^{-1}$

Subject to: $g_1(\mathbf{x}) = 4 - x_1^2 - x_2^2 \leq 0$

$$g_2(\mathbf{x}) = x_1^2 + x_2^2 - 16 \leq 0$$

$$h_1(\mathbf{x}) = x_1 - x_2 = 3$$

8.32

Minimize: $-x_1$

Subject to: $\exp(x_1^4 x_2) - 1 = 0$

$\exp(x_1^4 + x_2^4 - 1) = 1$

Starting point (nonfeasible): $\mathbf{x}^0 = [2.0, 1.0]^T$

Do you get both solutions?

$$\mathbf{x}^* = [\pm 1.0, 0.0]^T \quad \text{and} \quad [0.0, \pm 1.0]^T$$

8.33

Minimize: $-x_1$

Subject to: $x_1^5 + 3x_1^4 x_2^2 + 3x_1^2 x_2^4 + x_2^6 - 4x_1^4 + 8x_1^2 x_2^2 - 4x_2^4 = 0$

Starting point (nonfeasible): $\mathbf{x}^0 = [-2.0, 2.0]^T$

Do you get all three solutions?

$$\mathbf{x}^* = [2.0, 0.0]^T, \quad [0.0, 0.0]^T, \quad [0.0, 2.0]^T$$

8.34 The cost of constructing a distillation column can be written

$$C = C_p AN + C_s HAN + C_f + C_d + C_b + C_L + C_x \quad (\text{a})$$

where C = Total cost, \$

C_p = cost per square foot of plate area, \$/ft²

A = column cross-sectional area, ft²

N = number of plates

N_{\min} = minimum number of plates

C_s = cost of shell, \$/ft³

H = distance between plates, ft

C_f = cost of feed pump, \$

C_d = cost of distillate pump, \$

C_b = cost of bottoms pump, \$

C_L = cost of reflux pump, \$

C_x = other fixed costs, \$

The problem is to minimize the total cost, once produce specifications and the throughput are fixed and the product and feed pumping costs are fixed; that is, C_f , C_d , C_L , and C_b are fixed. After selection of the material of construction, the costs are determined; that is, C_p , C_s , C_x are also fixed.

The process variables can be related through two empirical equations:

$$\frac{L}{D} = \left[\frac{1}{1 - (N_{\min}/N)} \right]^x \left(\frac{L}{D} \right)_{\min} \quad (\text{b})$$

$$A = K(L + D)^\beta \quad (\text{c})$$

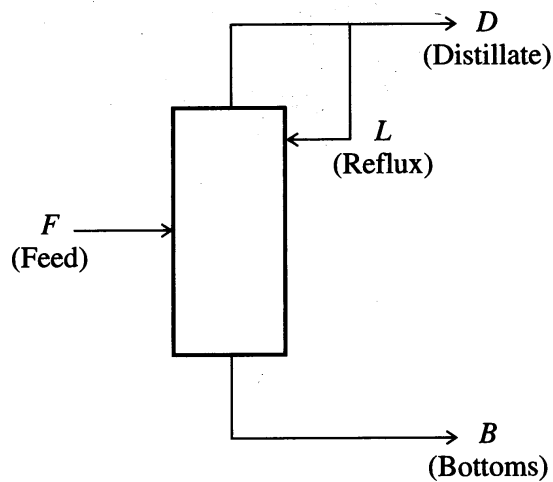


FIGURE P8.34

For simplicity choose $\alpha = \beta = 1$; then

$$\frac{L}{D} = \left[\frac{1}{1 - (N_{\min}/N)} \right] \left(\frac{L}{D} \right)_{\min} \quad (\text{b}')$$

$$A = K(L + D) \quad (\text{c}')$$

For a certain separation and distillation column the following parameters are known to apply:

$$C_p = 30 \quad C_x = 8000$$

$$C_s = 10 \quad F = 1500$$

$$H = 2 \quad D = 1000$$

$$C_f = 4000 \quad N_{\min} = 5$$

$$C_d = 3000 \quad \left(\frac{L}{D} \right)_{\min} = 1$$

$$C_b = 2000 \quad K = \frac{1 \text{ (h)(ft}^2\text{)}}{100 \text{ lb}}$$

The pump cost for the reflux stream can be expressed as

$$C_L = 5000 + 0.7L \quad (\text{d})$$

- Determine the process decision or independent variables. Which variables are dependent?
- Find the minimum total cost and corresponding values of the variables.

- 8.35** A chemical manufacturing company sells three products and has found that its revenue function is $f = 10x + 4.4y^2 + 2z$, where x , y , and z are the monthly production rates of each chemical. It is found from breakeven charts that it is necessary to impose the following limits on the production rates:

$$x \geq 2$$

$$\frac{1}{2}z^2 + y^2 \geq 3$$

In addition, only a limited amount of raw material is available; hence the following restrictions must be imposed on the production schedule:

$$x + 4y + 5z \leq 32$$

$$x + 3y + 2z \leq 29$$

Determine the best production schedule for this company, and find the best value of the revenue function.

- 8.36** A problem in chemical equilibrium is to minimize

$$f(\mathbf{x}) = \sum_{i=1}^n x_i \left(w_i + \ln P + \ln \frac{x_i}{\sum_{i=1}^n x_i} \right)$$

subject to the material balances

$$x_1 + 2x_2 + 2x_3 + x_6 + x_{10} = 2$$

$$x_4 + 2x_5 + x_6 + x_7 = 1$$

$$x_3 + x_7 + x_8 + 2x_9 + x_{10} = 1$$

Given $P = 750$ and w_i ,

i	w_i	i	w_i
1	-10.021	6	-18.918
2	-21.096	7	-28.032
3	-37.986	8	-14.640
4	-9.846	9	-30.594
5	-28.653	10	-26.111

what is \mathbf{x}^* and $f(\mathbf{x}^*)$?

- 8.37** The objective is to fit a fifth-order polynomial to the curve $y = x^{1/3}$. To avoid fluctuations from the desired curve, divide the curve into ten points.

$$\mathbf{x}_i (i = 1, \dots, 10) = (0.5, 1, 4.5, 8, 17.5, 27, 45.5, 64, 94.5, 125)$$

and fit the polynomial (find the values of a_i)

$$P(\mathbf{a}, \mathbf{x}) = a_1x + a_2x^2 + a_3x^3 + a_4x^4 + a_5x^5$$

by solving the following problem

$$\text{Minimize: } f(\mathbf{x}) \sum_{i=1}^{10} [P(\mathbf{a}, x_i) - x_i^{1/3}]^2$$

$$\text{Subject to: } 0 \leq P(\mathbf{a}, j) \leq 5, \quad j = 1, 8, 27, 64$$

$$P(\mathbf{a}, 125) = 5$$

8.38 The Williams–Otto process as posed in this problem involves ten variables and seven constraints leaving 3 degrees of freedom. Three starting points are shown in Table P8.38.1. Find the maximum Q and the values of the ten variables from one of the starting points (S.P.). The minimum is very flat.

Figure P8.38 shows a simplified block diagram of the process. The plant consists of a perfectly stirred reactor, a decanter, and a distillation column in series. There is recycle from the column reboiler to the reactor.

The mathematical descriptions of each plant unit are summarized in Tables P8.38.2 and P8.38.3. The return function for this process as proposed by Williams and Otto (1960) and slightly modified by DiBella and Stevens (1965) to a variable reactor volume problem is

$$\begin{aligned} \text{Maximize: } Q &= \frac{100}{600 V_\rho} [8400(0.3F_P + 0.0068 F_D - 0.02F_A - 0.03F_B - 0.01F_G) \\ &\quad - 2.22F_R - (0.124)(8400)(0.3F_P + 0.0068F_D) - 60V_\rho] \\ &= \text{Return (\%)} \end{aligned}$$

TABLE P8.38.1
Starting points

Variable	x	x_1	x_2	x_3
F_{RA}	1	18,187	6,000	22,381
F_{RB}	2	60,815	35,000	72,297
F_{RC}	3	3,331	10,000	4,391
F_{RE}	4	60,542	15,000	78,144
F_G	5	3,609	5,000	3,140
F_{RP}	6	10,817	6,000	12,557
α	7	0.761	0.789	0.81
F_A	8	13,546	10,000	12,876
F_B	9	31,523	40,000	29,416
T	10	656	610	648

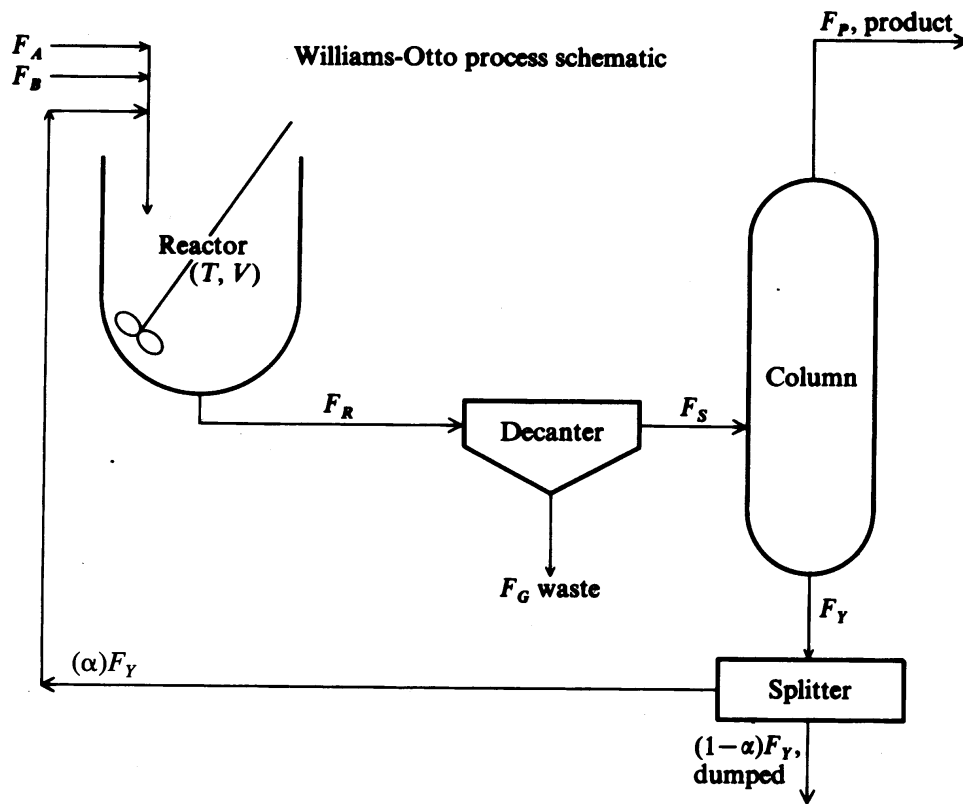


FIGURE P8.38

TABLE P8.38.2

g_i = residual of mass balance on component i , $i = A, B, C, E, G$

Constraints
$g_1 = F_A + F_{TA} - F_{RA} - P_1 = 0$
$g_2 = F_B + F_{TB} - F_{RB} - P_1 - P_2 = 0$
$g_3 = F_{TC} + 2P_1 - F_{RC} - 2P_2 - P_3 = 0$
$g_4 = F_{TE} + 2P_2 - F_{RE} = 0$
$g_5 = F_{TC} + P_2 - F_{RG} - 0.5P_3 = 0$
$g_6 =$ overall mass balance $= F_A + F_B - F_D - F_P = 0$
$g_7 =$ production requirement $= F_P - 4763 = 0$
$0 \leq \alpha \leq 1 \quad 500 \leq T \leq 1000$

TABLE P8.38.3
Williams–Otto unit mathematical models

Decanter			
$F_{Si} = F_{Ri}$	$i = A, B, C, E, P$		
$F_{SG} = 0$			
$F_{Cb} = F_{RC}$			
Distillation column			
$F_{Yi} = F_{Si}$	$i = A, B, C, E, G$		
$F_P = F_{SP} - 0.1F_{SE}$			
$F_{YP} = F_{SP} - F_P$			
Splitter			
$F_{Ti} = \alpha F_{Yi}$	$i = A, B, C, E, G, P$		
$F_{Di} = (1 - \alpha)F_{Yi}$			
Reactor ($V = 0.0002964 F_R$)	i	a_i	b_i
$K_i = a_i \exp(-b_i/T)V/F_R^2$			
$P_1 = K_1 F_{RA} F_{RB}$	1	5.9755×10^9	12,000
$P_2 = K_2 F_{RB} F_{RC}$	2	2.5962×10^{12}	15,000
$P_3 = K_3 F_{RC} F_{RG}$	3	9.6283×10^{15}	20,000

TABLE P8.38.4
Williams–Otto process nomenclature

F_A, F_B	Fresh feeds of components A, B (lb/h)
F_R	Total reactor output flow rate
F_{Ri}	Reactor output flow rate of component i
F_{Si}	Decanter output flow rate of component i
F_G	Decanter bottoms flow rate of component G
F_P	Column overhead flow rate of component P
F_{Yi}	Column bottoms flow rate of component i
F_D	Total column bottoms takeoff flow rate
F_{Di}	Column bottoms takeoff flow rate of component i
F_T	Total column bottoms recycle flow rate
F_{Ti}	Column bottoms recycle flow rate of component i
α	Fraction of column bottoms recycled to reactor
V	Reactor volume (ft ³)
ρ	Density of reaction mixture (assumed constant, 50 lb/ft ³)

8.39 Klein and Klimpel (1967) described an NLP involving the optimal selection of plant sites and plant sizes over time. The functions representing fixed and working capital were of the form

$$\text{Fixed capital: Cost} = a_0 + a_1 S^{a_2}$$

$$\text{Working capital: Cost} = b_0 + b_1 P + b_2 S^{a_2}$$

where S = plant size

P = annual production

a 's, b 's = known constants obtained empirically

Variable annual costs were expressed in the form of

$$\text{Cost} = P(c_1 + c_2 S + c_3 S^{c_4})$$

Transportation costs were assumed to be proportional to the size of the shipments for a given source and destination.

The objective function is the net present value, NPV (sum of the discounted cash flows), using a discount rate of 10 percent. All flows except capital were assumed to be uniformly distributed over the year; working capital was added or subtracted instantaneously at the beginning of each year, and fixed capital was added only in the zero year.

The continuous discounting factors were

1. For instantaneous funds,

$$F_i = e^{-ry} \quad (r = \text{interest rate, } y = \text{years hence})$$

2. For uniformly flowing funds,

$$F_u = \frac{e^r - 1}{r} e^{-ry}$$

The variable y may be positive (after year zero) or negative (before year zero) or zero (for year ending with point zero in time).

As prices and revenue were not considered, maximization of net present value was equivalent to minimization of net cost.

Let P_{ijk} be the amount of product shipped from location i ($i = 1, 2, 3, 4$) to market j ($j = 1, 2, 3$) in year k ($k = 0, 1, 2, 3$). Let S_i and \bar{S}_1 be, respectively, the size of plant in location i , and a variable restricted to 0 or 1, depending on whether S_i is 0. Furthermore, let M_{ojk} be the market demand at center j in year k . Finally, for the sake of convenience, let $P_{io k}$ denote the total production in plant i during year k .

The nonlinear programming problem is: Find S_i and P_{ijk} that will

$$\text{Maximize: } \sum_i \text{NPV} \quad (\text{including shipping})$$

$$\text{Subject to: } \sum_i P_{ijk} = M_{ojk}$$

$$S_i \geq 0, \quad P_{ijk} \geq 0$$

Table P8.39.1 indicates how the net present value was determined for location 1; NPV relations for the other locations were similarly formed. Table P8.39.2 lists the

TABLE P8.39.1
Net percent value

1. Contribution of fixed capital (plant 1)			
Year	Fixed capital	Discount factor	Discounted cash flow
0(1967)	$0.7\bar{S}_1 + 1.5S_1^6$	1.0517	$-0.7362\bar{S}_1 - 1.5775S_1^6$
2. Contribution of working capital (plant 1)			
Year end	Working capital	Discount factor	Discounted cash flow at 10% discount rate
0	$0.4\bar{S}_1 + 0.2P_{101} + 0.05S_1^6$	1.000	$-0.4\bar{S}_1 - 0.2P_{101} - 0.05S_1^6$
1	$0.2(P_{102} - P_{101})$	0.9048	$-0.1810P_{102} + 0.1810P_{101}$
2	$0.2(P_{103} - P_{102})$	0.8187	$-0.1637P_{103} + 0.1637P_{102}$
3	$-0.4\bar{S}_1 - 0.2P_{103} - 0.05S_1^6$	0.7408	$+0.2963\bar{S}_1 + 0.1482P_{103} + 0.0370S_1^6$
3. Contribution of operational cost (plant 1) a. Cost tabulation (excluding shipping)			
Year	Amount	Depreciation*	Other costs
1	P_{101}	$0.4667\bar{S}_1 + 1.0S_1^6$	$0.03\bar{S}_1 - 0.01S_1 + 0.05S_1^{0.45} + 0.07S_1^{0.6} + 0.1P_{101}$ $-0.05P_{101}S_1 + 0.4P_{101}S_1^{-0.55}$
2	P_{102}	$0.1167\bar{S}_1 + 0.25S_1^6$	$0.03S_1 - 0.01S_1 + 0.05S_1^{0.45} + 0.07S_1^{0.6} + 0.095P_{102}$ $-0.0048P_{102}S_1 + 0.38P_{102}S_1^{-0.55}$
3	P_{103}	$0.1166\bar{S}_1 + 0.25S_1^6$	$0.03S_1 - 0.01S_1 + 0.05S_1^{0.45} + 0.07S_1^{0.6} + 0.0903P_{103}$ $-0.0045P_{103}S_1 + 0.361P_{103}S_1$
b. Discounted cash flow of costs (plant 1)			
Year	Discount factor	Discounted cost flow at 10% discount rate	
1	0.9516	$0.1983S_1 + 0.0049\bar{S}_1 - 0.0247S_1^{0.45} + 0.4221S_1^{0.6} - 0.0495P_{101}$ $+ 0.0025P_{101}S_1 - 0.1979P_{101}S_1^{-0.55}$	
2	0.8611	$0.0348S_1 + 0.0045\bar{S}_1 - 0.0224S_1^{0.45} + 0.0720S_1^{0.6} - 0.0425P_{102}$ $+ 0.0020P_{102}S_1 - 0.1702P_{102}S_1^{-0.55}$	
3	0.7791	$0.0315S_1 + 0.0041\bar{S}_1 - 0.0203S_1^{0.45} + 0.0651S_1^{0.6} - 0.0366P_{103}$ $+ 0.0017P_{103}S_1 - 0.1463P_{103}S_1^{0.55}$	
4. Contribution of shipping costs (from plant 1)			
Year	Discount factor	Shipping cost	Discounted cash flow at 10% discount rate
1	0.9516	$0.8P_{121} + 0.5P_{121}$	$-0.396P_{121} - 0.247P_{131}$
2	0.8611	$0.7P_{122} + 0.45P_{132}$	$-0.313P_{122} - 0.201P_{132}$
3	0.7791	$0.6P_{123} + 0.4P_{133}$	$-0.243P_{123} - 0.162P_{133}$

*Method of double rate-declining balance and straight-line crossover was used.

TABLE P8.39.2
The objective function

$$\begin{aligned}
 Z_{\max} = & -0.5753\bar{S}_1 - 1.0313S_1^{0.6} - 0.0685P_{101} - 0.0597P_{102} - 0.0522P_{103} + 0.0135S_1 \\
 & - 0.0674S_1^{0.45} + 0.0025P_{101}S_1 + 0.0020P_{102}S_1 + 0.0017P_{103}S_1 \\
 & - 0.1979P_{101}S_1^{-0.55} - 0.1702P_{102}S_1^{-0.55} - 0.1463P_{103}S_1^{0.55} - 0.396P_{121} \\
 & - 0.247P_{131} - 0.313P_{122} - 0.202P_{132} - 0.243P_{123} - 0.162P_{133} - 0.3428\bar{S}_2 \\
 & - 0.8920S_2^{0.6} - 0.0685P_{201} - 0.0597P_{202} - 0.0522P_{203} + 0.0135S_2 - 0.0809S_1^{0.45} \\
 & + 0.0025P_{201}S_2 + 0.0020P_{202}S_2 + 0.0017P_{203}S_2 - 0.02227P_{201}S_2^{-0.55} \\
 & - 0.1914P_{202}S_2^{-0.55} - 0.1645P_{203}S_2^{-0.55} - 0.396P_{211} - 0.495P_{231} - 0.313P_{212} \\
 & - 0.448P_{232} - 0.243P_{213} - 0.405P_{233} - 0.3164\bar{S}_3 - 1.2987S_3^{0.6} - 0.0942P_{301} \\
 & - 0.0819P_{302} - 0.0712P_{303} - 0.0539S_3^{0.45} + 0.0030P_{301}S_3 + 0.0026P_{302}S_3 \\
 & + 0.0022P_{303}S_3 - 0.2227P_{301}S_3^{-0.55} - 0.1914P_{302}S_3^{-0.55} - 0.1645P_{303}S_3^{0.55} \\
 & - 0.247P_{311} - 0.495P_{321} - 0.202P_{312} - 0.448P_{322} - 0.162P_{313} - 0.405P_{323} \\
 & - 0.2441\bar{S}_4 - 1.3707S_4^{0.6} - 0.0577P_{401} - 0.0504P_{402} - 0.0440P_{403} \\
 & + 0.0020P_{401}S_4 + 0.0017P_{402}S_4 + 0.0015P_{403}S_4 - 0.1484P_{401}S_4^{-0.55} \\
 & - 0.1276P_{402}S_4^{-0.55} - 0.1097P_{403}S_4^{-0.55} - 0.495P_{411} - 0.099P_{421} - 0.040P_{431} \\
 & - 0.448P_{412} - 0.090P_{422} - 0.040P_{432} - 0.405P_{413} - 0.088P_{423} - 0.041P_{433}
 \end{aligned}$$

TABLE P8.39.3
The constants

(1) $S_1 + S_2 + S_3 + S_4 = 10$	(2) $P_{111} + P_{211} + P_{311} + P_{411} = 1$	
(3) $P_{112} + P_{212} + P_{312} + P_{412} = 4$	(4) $P_{113} + P_{213} + P_{313} + P_{413} = 5$	
(5) $P_{121} + P_{221} + P_{321} + P_{421} = 2$	(6) $P_{122} + P_{222} + P_{322} + P_{422} = 3$	
(7) $P_{123} + P_{223} + P_{323} + P_{423} = 2$	(8) $P_{131} + P_{231} + P_{331} + P_{431} = 4$	
(9) $P_{132} + P_{232} + P_{332} + P_{432} = 3$	(10) $P_{133} + P_{233} + P_{333} + P_{433} = 2$	
(11) $P_{101} - S_2 \leq 0$	(12) $P_{102} - S_1 \leq 0$	(13) $P_{103} - S_1 \leq 0$
(14) $P_{201} - S_2 \leq 0$	(15) $P_{202} - S_2 \leq 0$	(16) $P_{203} - S_2 \leq 0$
(17) $P_{301} - S_3 \leq 0$	(18) $P_{302} - S_3 \leq 0$	(19) $P_{303} - S_2 \leq 0$
(20) $P_{401} - S_4 \leq 0$	(21) $P_{402} - S_4 \leq 0$	(22) $P_{403} - S_4 \leq 0$

overall objective function, and Table P8.39.3 lists (1) the 22 constraints, (2) one equation constraining the total plant capacity to be 10 million pounds per year, (3) nine equations requiring satisfaction of the three markets every year, and (4) 12 inequalities calling for plant production not to exceed plant capacity. In addition, the nonnegativity constraints are applicable to all 40 variables. Thus the problem has 10 linear equality constraints and 52 inequality constraints.

- 8.40** Consider the problem of minimizing the purchase of fuel oil when it is needed to produce an output of 50 MW from a two-boiler turbine-generator combination that can use fuel oil or blast furnace gas (BFG) or any combination of these. The maximum available BFG is specified.

By applying nonlinear curve fitting, we obtained the fuel requirements for the two generators explicitly in terms of MW produced. For generator 1 we have the fuel requirements for fuel oil in tons per hour (x_{11})

$$f_1 = 1.4609 + 0.15186x_{11} + 0.00145x_{11}^2$$

and for BFG in fuel units per hour (x_{12})

$$f_2 = 1.5742 + 0.1631x_{12} + 0.001358x_{12}^2$$

where ($x_{11} + x_{12}$) is the output in MW of generator 1. The range of operation of the generator is

$$18 \leq (x_{11} + x_{12}) \leq 30$$

Similarly for generator 2 the requirement for fuel oil is

$$g_1 = 0.8008 + 0.2031x_{21} + 0.000916x_{21}^2$$

and for BFG,

$$g_2 = 0.7266 + 0.2256x_{22} + 0.000778x_{22}^2$$

where ($x_{21} + x_{22}$) is the output in MW of generator 2. The range of operation of the second generator is

$$14 \leq (x_{21} + x_{22}) \leq 25$$

It is assumed that only 10.0 fuel units of BFG are available each hour and that each generator may use any combination of fuel oil or BFG. It is further assumed that when a combination of fuel oil and BFG is used, the effects are additive.

The problem is to produce 50 MW from the two generators in such a way that the amount of fuel oil consumed is minimum. Use successive linear programming.

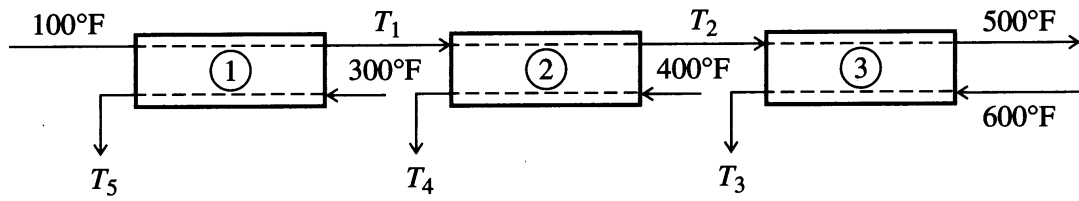
- 8.41** For the purposes of planning you are asked to determine the optimal heat exchanger areas for the sequence of three exchangers as shown in Figure P8.41.

Data:

Exchanger	Overall heat transfer coefficient [U Btu/(h)(ft ²)(°F)]	Area required (ft ²)	Duty (Btu/h)
1	$U_1 = 120$	A_1	Q_1
2	$U_2 = 80$	A_2	Q_2
3	$U_3 = 40$	A_3	Q_3

$$wCp = 10^5 \text{ Btu/(h)(°F)}$$

Hint: Find the temperatures T_1, T_2, T_3 such that $\sum A_i$ is a minimum.

**FIGURE P8.41**

MIXED-INTEGER PROGRAMMING

9.1 Problem Formulation	352
9.2 Branch-and-Bound Methods Using LP Relaxations	354
9.3 Solving MINLP Problems Using Branch-and-Bound Methods	361
9.4 Solving MINLPs Using Outer Approximation	369
9.5 Other Decomposition Approaches for MINLP	370
9.6 Disjunctive Programming	371
References	372
Supplementary References	373
Problems	374

Introduction

Many problems in plant operation, design, location, and scheduling involve variables that are not continuous but instead have integer values. Decision variables for which the levels are a dichotomy—to install or not install a new piece of equipment, for example—are termed “0–1” or binary variables. Other integer variables might be real numbers 0, 1, 2, 3, and so on. Sometimes we can treat integer variables as if they were continuous, especially when the range of a variable contains a large number of integers, such as 100 trays in a distillation column, and round the optimal solution to the nearest integer value. Although this procedure leads to a suboptimal solution, the solution is quite acceptable from a practical viewpoint. However, for a small range of a variable such as 1 to 3, when the optimal solution yields a value of 1.3, we have less confidence in rounding. In this section we will illustrate some examples of problem formulation and subsequent solution in which one or more variables are treated as integer variables.

First let us classify the types of problems that are encountered in optimization with discrete variables. The most general case is a *mixed integer programming* (MIP) problem in which the objective function depends on two sets of variables, \mathbf{x} and \mathbf{y} ; \mathbf{x} is a vector of continuous variables and \mathbf{y} is a vector of integer variables. A problem involving only integer variables is classified as an *integer programming* (IP) problem. Finally, a special case of IP is *binary integer programming* (BIP), in which all of the variables \mathbf{y} are either 0 or 1. Many MIP problems are linear in the objective function and constraints and hence are subject to solution by linear programming. These problems are called *mixed-integer linear programming* (MILP) problems. Problems involving discrete variables in which some of the functions are nonlinear are called *mixed-integer nonlinear programming* (MINLP) problems. We consider both linear and nonlinear MIP problems in this chapter.

9.1 PROBLEM FORMULATION

Here we review some classical formulations of typical integer programming problems that have been discussed in the operations research literature, as well as some problems that have direct applicability to chemical processing:

1. *The knapsack problem.* We have n objects. The weight of the i th object is w_i , and its value is v_i . Select a subset of the objects such that their total weight does not exceed W (the capacity of the knapsack) and their total value is a maximum.

$$\text{Maximize: } f(\mathbf{y}) = \sum_{i=1}^n v_i y_i$$

$$\text{Subject to: } \sum_{i=1}^n w_i y_i \leq W \quad y_i = 0, 1 \quad i = 1, 2, \dots, n$$

The binary variable y_i indicates whether an object i is selected ($y_i = 1$) or not selected ($y_i = 0$).

2. *The traveling salesman problem.* The problem is to assign values of 0 or 1 to variables y_{ij} , where y_{ij} is 1 if the salesman travels from city i to city j and 0 otherwise. The constraints in the problem are that the salesman must start at a particular city, visit each of the other cities only once, and return to the original city. A cost (here it is distance) c_{ij} is associated with traveling from city i to city j , and the objective function is to minimize the total cost of the trips to each city visited, that is

$$f(\mathbf{y}) = \sum_{i=1}^n \sum_{j=1}^n c_{ij} y_{ij}$$

subject to the $2n$ constraints

$$\sum_{i=1}^n y_{ij} = 1, \quad \sum_{j=1}^n y_{ij} = 1 \quad \begin{array}{ll} y_{ij} = 0, 1 & i, j = 1, \dots, n \\ y_{ij} = 0 & i = j \end{array}$$

The two types of equality constraints ensure that each city is only visited once in any direction. We define $y_{ii} = 0$ because no trip is involved. The equality constraints (the summations) ensure that each city is entered and exited exactly once. These are the constraints of an assignment problem (see Section 7.8). In addition, constraints must be added to ensure that the y_{ij} which are set equal to 1 correspond to a single circular tour or cycle, not to two or more disjoint cycles. For more information on how to write such constraints, see Nemhauser and Wolsey (1988).

For a chemical plant analogy, the problem can also be cast in terms of processing n batches on a single piece of equipment in which the equipment is reset between processing the i th and j th batches. The batches can be processed in any order. Here, c_{ij} is the time or cost required to “set up” the equipment to do batch j if it was previously doing batch i , and $y_{ij} = 1$ means batch i is immediately followed by batch j .

3. *Blending problem.* You are given a list of possible ingredients to be blended into a product, from a list containing the weight, value, cost, and analysis of each ingredient. The objective is to select from the list a set of ingredients so as to have a satisfactory total weight and analysis at minimum cost for a blend. Let x_j be the quantity of ingredient j available in continuous amounts and y_k represent ingredients to be used in discrete quantities v_k ($y_k = 1$ if used and $y_k = 0$ if not used). Let c_j and d_k be the respective costs of the ingredients and a_{ij} be the fraction of component i in ingredients j . The problem statement is

$$\text{Minimize: } \sum_j c_j x_j + \sum_k d_k v_k y_k$$

$$\text{Subject to: } W^l \leq \sum_j x_j + \sum_k v_k y_k \leq W^u$$

$$A_i^l \leq \sum_j a_{ij} x_j + \sum_k a_{ik} v_k y_k \leq A_i^u$$

$$0 \leq x_j \leq u_j \quad \text{for all } j$$

$$y_k = (0, 1) \quad \text{for all } k$$

where u_j = upper limit of the j th ingredient,

W^l and W^u = the lower and upper bounds on the weights, respectively

A_i^l and A_i^u = the lower and upper bounds on the analysis for component i , respectively

4. *Location of oil wells (plant location problem)*. It is assumed that a specific production–demand versus time relation exists for a reservoir. Several sites for new wells have been designated. The problem is how to select from among the well sites the number of wells to be drilled, their locations, and the production rates from the wells so that the difference between the production–demand curve and flow curve actually obtained is minimized. Refer to Rosenwald and Green (1974) and Murray and Edgar (1978) for a mathematical formulation of the problem. The integer variables are the drilling decisions (0 = not drilled, 1 = drilled) for a set of n possible drilling locations. The continuous variables are the different well production rates. This problem is related to the plant location problem and also the *fixed-charge problem* (Hillier and Lieberman, 1986).

Many other problems can be formulated as integer programming problems; refer to the examples in this chapter and Nemhauser and Wolsey (1988) and the supplementary references for additional examples.

Integer and mixed-integer programs are much harder to solve than linear programs. The computation time of even the best available MIP solvers often increases rapidly with the number of integer variables, although this effect is highly problem-dependent. This is partially caused by the exponential increase in the total number of possible solutions with problem size. For example, a traveling salesman problem with n cities has $n!$ tours, and there are 2^n solutions to a problem with n binary variables (some of which may be infeasible).

In this chapter, we discuss solution approaches for MILP and MINLP that are capable of finding an optimal solution and verify that they have done so. Specifically, we consider branch-and-bound (BB) and outer linearization (OL) methods. BB can be applied to both linear and nonlinear problems, but OL is used for nonlinear problems by solving a sequence of MILPs. Chapter 10 further considers branch-and-bound methods, and also describes heuristic methods, which often find very good solutions but are unable to verify optimality.

9.2 BRANCH-AND-BOUND METHODS USING LP RELAXATIONS

Branch and bound (BB) is a class of methods for linear and nonlinear mixed-integer programming. If carried to completion, it is guaranteed to find an optimal solution to linear and convex nonlinear problems. It is the most popular approach and is currently used in virtually all commercial MILP software (see Chapter 7).

Consider the application of BB to a general MILP problem, in which all the integer variables are binary, that is, either 0 or 1. The problem formed by relaxing the “0 or 1” constraint to “anywhere between 0 and 1” is called the LP relaxation of the MILP. BB starts by solving this LP relaxation. If all discrete variables have integer values, this solution solves the MILP. If not, one or more discrete variables has a fractional value. BB chooses one of these variables in its *branching* step and then creates two LP subproblems by fixing this variable first at 0, then at 1. If either of these subproblems has an integer solution, it need not be investigated further. If its objective value is better than the best value found thus far, it replaces this best value. If either subproblem is infeasible, it need not be investigated further. Otherwise, we find another fractional variable and repeat the steps. A clever *bounding* test can also be applied to each subproblem. If the test is satisfied, the subproblem need not be investigated further. This bounding test, together with the rest of the procedure, is explained in the following example.

EXAMPLE 9.1 BRANCH-AND-BOUND ANALYSIS OF AN INTEGER LINEAR PROGRAM

$$\text{Maximize: } f = 86y_1 + 4y_2 + 40y_3$$

$$\text{Subject to: } 774y_1 + 76y_2 + 42y_3 \leq 875$$

$$67y_1 + 27y_2 + 53y_3 \leq 875$$

$$y_1, y_2, y_3 = 0, 1$$

We can show the various subproblems developed from the stated problem by a tree (Figure E9.1). The objective function and inequality constraints are the same for each subproblem and so are not shown. The upper bound and lower bound for f are represented by ub and lb , respectively.

Each subproblem corresponds to a node in the tree and represents a *relaxation* of the original IP. One or more of the integer constraints $y_i = 0$ or 1 are replaced by the *relaxed* condition $0 \leq y_i \leq 1$, which includes the original integers, but also all of the real values in between.

Node 1. The first step is to set up and solve the relaxation of the binary IP via LP. The optimal solution has one fractional (noninteger) variable (y_2) and an objective function value of 129.1. Because the feasible region of the relaxed problem includes the feasible region of the initial IP problem, 129.1 is an *upper bound* on the value of the objective function of the IP. If we knew a feasible binary solution, its objective value would be a *lower bound* on the value of the objective function, but none is assumed here, so the lower bound is set to $-\infty$. There is as yet no *incumbent*, which is the best feasible integer solution found thus far.

At node 1, y_2 is the only fractional variable, and hence any feasible integer solution must satisfy either $y_2 = 0$ or $y_2 = 1$. We create two new relaxations represented by nodes 2 and 3 by imposing these two integer constraints. The process of creating these two relaxed subproblems is called *branching*. The feasible regions of these two LPs are

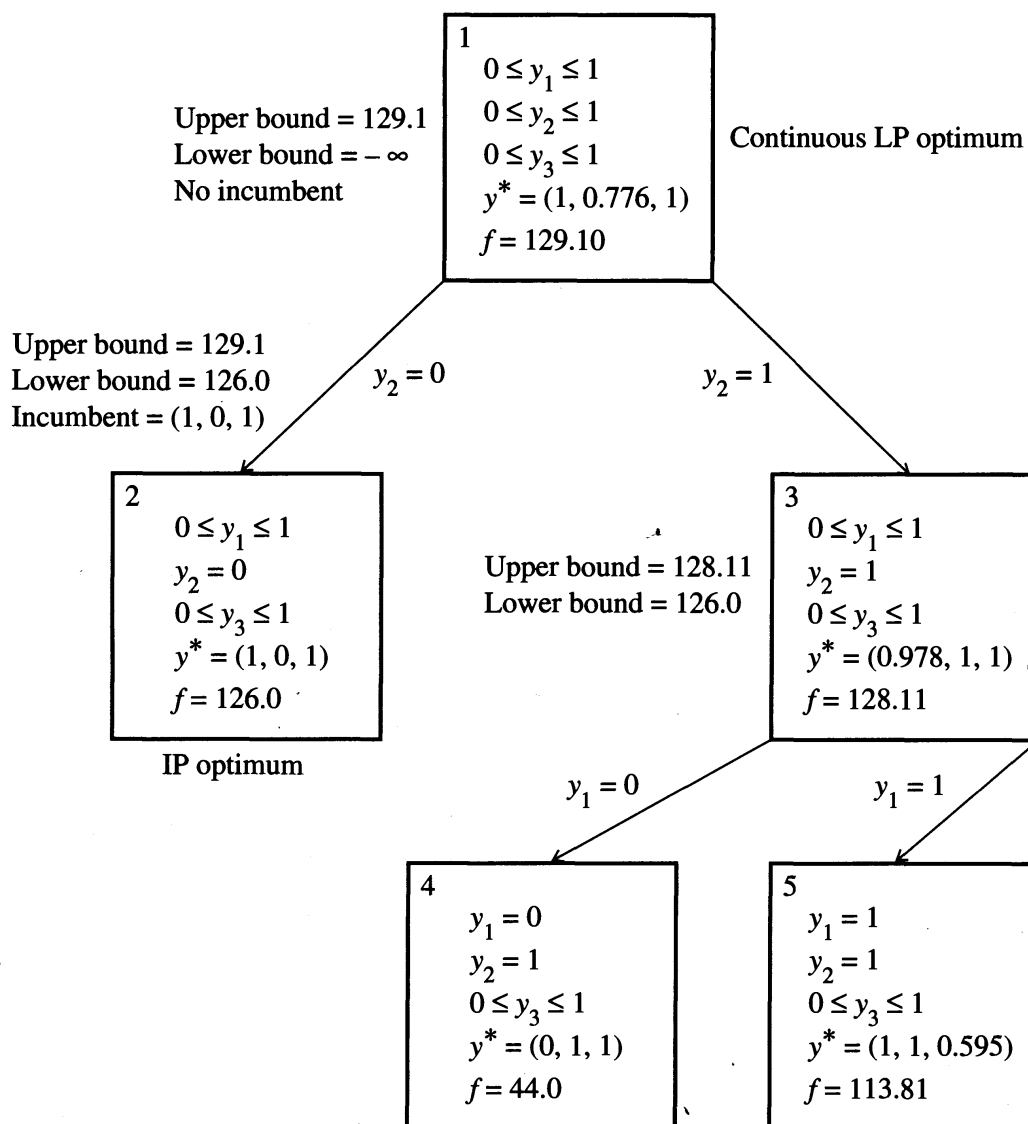


FIGURE E9.1

Decomposition of Example 9.1 via the branch-and-bound method.

partitions of the feasible region of the original IP, and one (or both) contain an optimal integer solution, if one exists (the problem may not have a feasible integer solution).

If the relaxed IP problem at a given node has an optimal binary solution, that solution solves the IP, and there is no need to proceed further. This node is said to be *fathomed*, because we do not need to branch from it. If a relaxed LP problem has several fractional values in the solution, you must select one of them to branch on. It is important to make a good choice. Branching rules have been studied extensively (see Nemhauser and Wolsey, 1988). Finally, if the node 1 problem has no feasible solution, the original IP is infeasible. At this point, the two nodes resulting from branching are unfathomed, and you must decide which to process next. How to make the decision has been well studied (Nemhauser and Wolsey, 1988, Chapter II.4).

Node 2. For this example we choose node 2 and find that the solution to the relaxed problem is a binary solution, so this node is now fathomed. The solution is the

first feasible integer solution found, so its objective value of 126.0 becomes the current lower bound. The difference ($ub - lb$) is called the “gap,” and its value at this stage is $129.1 - 126.0 = 3.1$. It is common to terminate the BB algorithm when

$$\frac{\text{Gap}}{1.0 + |lb|} \leq \text{tol} \quad (9.1)$$

When the gap is smaller than some fraction tol of the incumbent’s objective value (the factor 1.0 ensures that the test makes sense when $lb = 0$). When $lb = -\infty$, you will always satisfy Equation 9.1. A tol value of 10^{-4} would be a tight tolerance, 0.01 would be neither tight nor loose, and 0.03 or higher would be loose. The termination criterion used in the Microsoft Excel Solver has a default tol value of 0.05.

Node 3. The solution of the problem displayed in node 3 is fractional with a value of the objective function equal to 128.11, so the upper bound for this node and all its successors is 128.11. The gap is now 2.11, so $\text{gap}/[1 + \text{abs}(lb)] = 0.0166$. If tol in Equation (9.1) is larger than this, the BB algorithm stops. Otherwise, we create two new nodes by branching on y_1 .

Node 4. Node 4 has an integer solution, with an objective function value of 44, which is smaller than that of the incumbent obtained previously. The incumbent is unchanged, and this node is fathomed.

Node 5. Node 5 has a fractional solution with an objective function value of 113.81, which is smaller than the lower bound of 126.0. Any successors of this node have objective values less than or equal to 113.81 because their LP relaxations are formed by adding constraints to the current one. Hence we can never find an integer solution with objective value higher than 126.0 by further branching from node 5, so node 5 is fathomed. Because there are no dangling nodes, the problem is solved, with the optimum corresponding to node 2.

EXAMPLE 9.2 BLENDING PRODUCTS INCLUDING DISCRETE BATCH SIZES

In this example we have two production units in a plant designated number 1 and number 2, making products 1 and 2, respectively, from the three feedstocks as shown in Figure E9.2a. Unit 1 has a maximum capacity of 8000 lb/day, and unit 2 of 10,000 lb/day. To make 1.0 lb of product 1 requires 0.4 lb of A and 0.6 lb of B; to make 1.0 lb of product 2 requires 0.3 lb of B and 0.7 lb of C. A maximum of 6000 lb/day of B is available, but there are no limits on the available amounts of A and C. Assume the net revenue after expenses from the manufacture of product 1 is \$0.16/lb, and of product 2 is \$0.20/lb. How much of products 1 and 2 should be produced per day, assuming that each must be made in batches of 2000 lb?

This problem is best formulated by scaling the production variables x_1 and x_2 to be in thousands of pounds per day, and the objective function to have values in thousands of dollars per day. This step ensures that all variables have values between 0 and 10 and often leads to both faster solutions and more readable reports. We formulate this problem as the following mixed-integer linear programming problem:

$$\text{Maximize: } f = 0.16x_1 + 0.2x_2$$

$$\text{Subject to: } x_i = 2y_i \quad i = 1, 2 \quad (a)$$

$$0.6x_1 + 0.3x_2 \leq 6 \quad (b)$$

$$0 \leq y_1 \leq 4 \quad 0 \leq y_2 \leq 5 \quad y_i \text{ integer} \quad (c)$$

Constraints (a) ensure that the scaled production amounts are even integers because the y_i are general integers subject to the bounds (c). The bounds on x_i are also implied by (a) and (c), and the x_i need not be declared an integer because they will be an integer if the y_i are.

A BB tree for this problem is in Figure E9.2b. The numbers to the left of each node are the current upper and lower bounds on the objective function, and the values to the right are the (y_1, y_2) values in the optimal solution to the LP relaxation at the node. The solution at node 1 has y_1 fractional, so we branch on y_1 , leading to nodes 2 and 3. If node 2 is evaluated first, its solution is an integer, so the node is fathomed, and $(2, 5)$ becomes the incumbent solution. This solution is optimal, but we do not

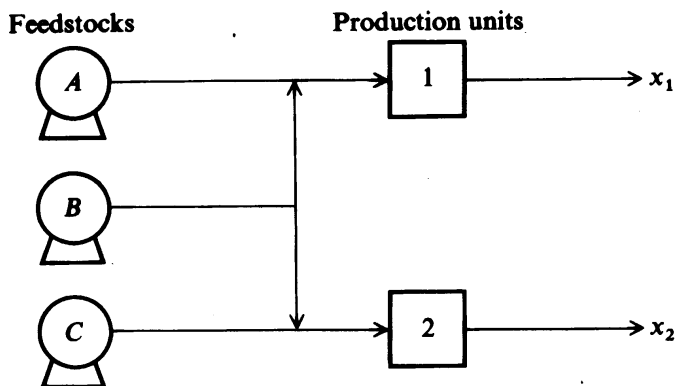


FIGURE E9.2a
Flow chart of a batch plant.

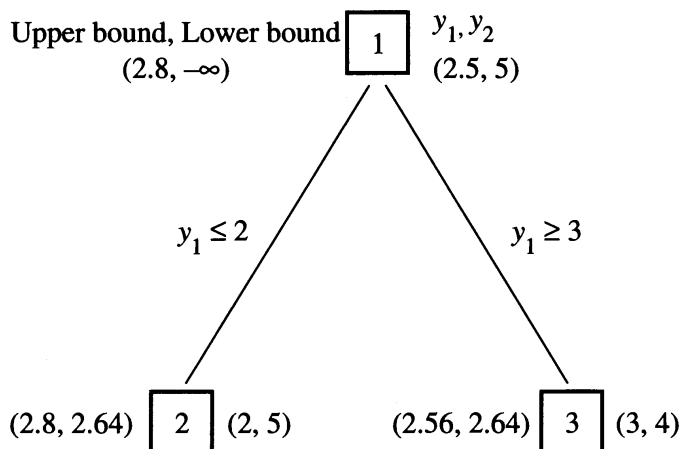


FIGURE E9.2b
Branch-and-bound tree.

	A	B	C	D	E	F	G	H
1	Example 9.2							
2	DECISION VARIABLES							
3								
4			x1	x2	y1	y2		
5		values	4	10	2	5		
6		bounds	8	10	4	5		
7	CONSTRAINTS							
8								
9							value	rhs
10		$x1-2y1=0$	1		-2		0	0
11		$x2-2y2=0$		1		-2	0	0
12		$.6x1+.3x2\leq 6$	0.6	0.3			5.4	6
13								
14	OBJECTIVE							
15								
16		$\max .16x1+.2x2$	0.16	0.2			2.64	

FIGURE E9.2c

Excel formulation for Example 9.2. Permission by Microsoft.

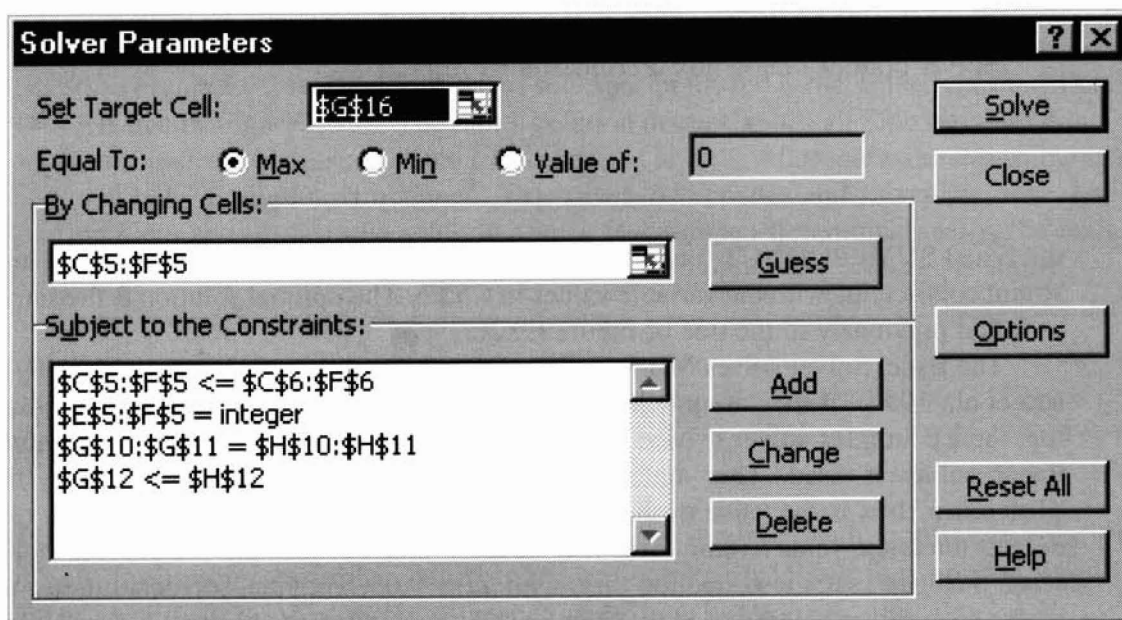


FIGURE E9.2d

Solver dialog for Example 9.2. Permission by Microsoft.

know that yet. Evaluating node 3, its solution is also an integer, so it is fathomed. Its solution has an objective function value of 2.56, smaller than the incumbent, so (2, 5) has been *proven* optimal. It is possible for a BB algorithm to discover an optimal solution at an early stage, but it may take many more steps to prove that it is optimal.

An Excel spreadsheet formulation of this problem is shown in Figures E9.2c and E9.2d. The constraint coefficient matrix is in the range C10:F12 and G10:G12 contains formulas that compute the values of the constraint functions. These formulas use

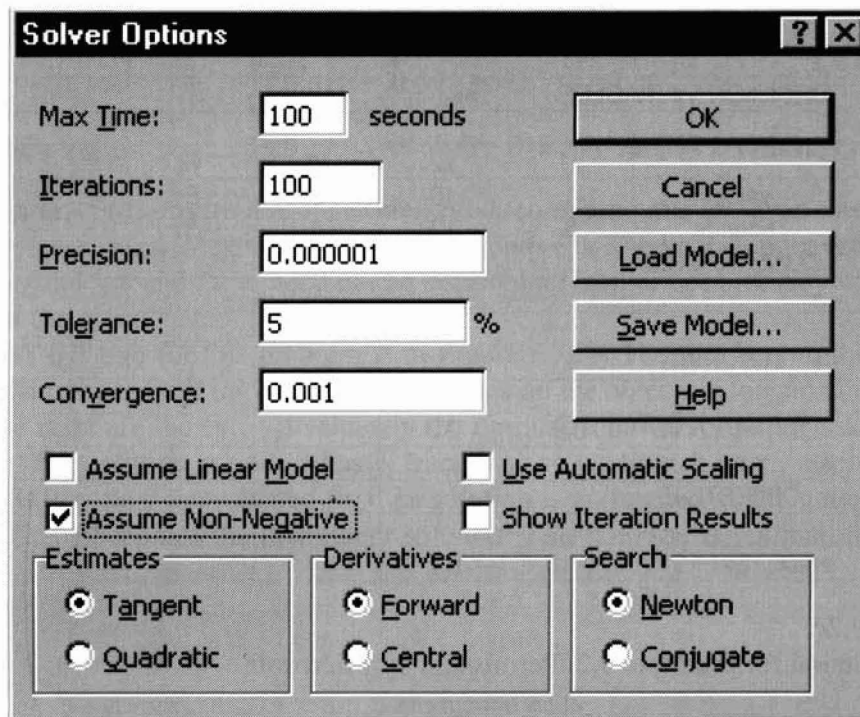


FIGURE E9.2e

Solver options dialog box. Permission by Microsoft.

the Excel SUMPRODUCT function to compute the inner product of the row of constraint coefficients with the variable values in C5:F5. The optimal solution is the same as found previously in the tree of Figure E9.2b.

The Excel Solver solves MILP and MINLP problems using a BB algorithm (Fylstra et al., 1998). If the “assume linear model” box is checked in the OPTIONS dialog, the LP simplex solver is used to solve the LP relaxations; if not, the GRG2 non-linear solver is used. This dialog is shown in Figure E9.2e. The value in the “Tolerance” box is the value of the tol in Equation (9.1). As shown in the figure, the default tolerance value is 0.05. This is a “loose” value because the BB process stops when the “gap” satisfies Equation (9.1) with $tol = 0.05$. The final solver solution can have an objective value that is as much as 5% worse than the optimal value. Users who are unaware of the meaning of the tolerance setting often assume that this final solution is optimal. For problems with few integer variables, you can safely use a tighter tolerance, for example, 0.1%, because such problems are usually solved quickly. For larger problems (e.g., more than 20 binary or integer variables), you can solve first with a loose tolerance. If this effort succeeds quickly, try again with a smaller tolerance.

If you request a sensitivity report after the solver has solved this example, the message “Sensitivity report and limits report are not meaningful for problems with integer constraints” appears (try it and see). A sensitivity report is “not meaningful” for a mixed-integer problem because Lagrange multipliers may not exist for such problems. To see why, recall that, in a problem with no integer variables, the Lagrange multiplier for a constraint is the derivative of the optimal objective value (OV) within

the *OV*. In other words, the *OV* function may not be differentiable at some points. As an example, consider the constraint

$$0.6x_1 + 0.3x_2 \leq 6$$

As shown in Figure E9.2c, this constraint is not active at the optimal solution because its left-hand side value is 5.4. Hence if its right-hand side is changed from 6 to 5.4, the optimal solution is unchanged. Now decrease this right-hand side (*RHS*) just a tiny bit further, to 5.3999). The new optimal objective value (*OV*) is 2.32, sharply worse than the *OV* of 2.64 when the *RHS* is 5.4. This *OV* change occurs because the small *RHS* decrease does not allow both x_1 and x_2 to retain their current values of 4 and 10, respectively. One or both must decrease, and because both are even integers, each must decrease by a value of 2. A small fractional change is not possible. The best possible change is to have $x_1 = 2$ while x_2 remains at 10. The ratio of *OV* change to *RHS* change is

$$\frac{\Delta OV}{\Delta RHS} = \frac{-0.32}{-0.0001} = 3200$$

Clearly as ΔRHS approaches zero the limit of this ratio does not exist; the ratio approaches infinity because ΔOV remains -0.32 . Hence the function *OV* (*RHS*) is not differentiable at $RHS = 5.4$, so no Lagrange multiplier exists at this point.

We now ask the reader to start Excel, either construct or open this model, and solve it after checking the “Show Iteration Results” box in the Solver Options dialog (see Figure E9.2d). The sequence of solutions produced is the same as is shown in the BB tree of Figure E9.2b. The initial solution displayed has all four variables equal to zero, indicating the start of the LP solution at node 1. After a few iterations, the optimal node 1 solution is obtained. The solver then creates and solves the node 2 subproblem and displays its solution after a few simplex iterations. Finally, the node 3 subproblem is created and solved, after which an optimality message is shown.

9.3 SOLVING MINLP PROBLEMS USING BRANCH-AND-BOUND METHODS

Many problems in plant design and operation involve both nonlinear relations among continuous variables, and binary or integer variables that appear linearly. The continuous variables typically represent flows or process operating conditions, and the binary variables are usually introduced for yes–no decisions. Such problems can be written in the following general form:

$$\text{Minimize: } z = f(\mathbf{x}) + \mathbf{c}^T \mathbf{y} \quad (9.2)$$

$$\text{Subject to: } \mathbf{h}(\mathbf{x}) = \mathbf{0} \quad (9.3)$$

$$\mathbf{g}(\mathbf{x}) + \mathbf{M}\mathbf{y} \leq \mathbf{0} \quad (9.4)$$

$$\mathbf{x} \in X, \quad \mathbf{y} \in Y \quad (9.5)$$

where \mathbf{x} is the vector of continuous variables, \mathbf{y} is the vector of integer (usually binary) variables, \mathbf{M} is a matrix, and X and Y are sets. The \mathbf{y} 's are typically chosen

to control the continuous variables \mathbf{x} by either forcing one (or more) variables to be zero or by allowing them to assume positive values. The choice of \mathbf{y} should be done in such a way that \mathbf{y} appears linearly, because then the problem is much easier to solve. The constraints (9.3) represent mass and energy balances, process input–output transformations, and so forth. The inequalities (9.4) are formulated so that \mathbf{y} influences \mathbf{x} in the desired way—we illustrate how to do this in several examples that follow. The set X is specified by bounds and other inequalities involving \mathbf{x} only, whereas Y is defined by conditions that the components of \mathbf{y} be binary or integer, plus other inequalities or equations involving \mathbf{y} only.

As discussed in Section 9.2, the Excel Solver uses a BB algorithm to solve MILP problems. It uses the same method to solve MINLP problems. The only difference is that for MINLP problems the relaxed subproblems at the nodes of the BB tree are continuous variable NLPs and must be solved by an NLP method. The Excel Solver uses the GRG2 code to solve these NLPs. GRG2 implements a GRG algorithm, as described in Chapter 8.

BB methods are guaranteed to solve either linear or nonlinear problems if allowed to continue until the “gap” reaches zero [see Equation (9.1)], provided that a global solution is found for each relaxed subproblem at each node of the BB tree. A global optimum can always be found for MILPs because both simplex and interior point LP solvers find global solutions to LPs because LPs are convex programming problems. In MINLP, if each relaxed subproblem is smooth and convex, then every local solution is a global optimum, and for these conditions many NLP algorithms guarantee convergence to a global solution.

Sufficient conditions on the functions in the general MINLP in Equations (9.2)–(9.5) to guarantee convexity of each relaxed subproblem are

1. The objective term $f(\mathbf{x})$ is convex.
2. Each component of the vector of equality constraint functions $\mathbf{h}(\mathbf{x})$ is linear.
3. Each component of the vector of inequality constraint functions $\mathbf{g}(\mathbf{x})$ is convex over the set X .
4. The set X is convex.
5. The set Y is determined by linear constraints and the integer restrictions on \mathbf{y} .

If these conditions hold, and an arbitrary subset of \mathbf{y} variables are fixed at integer values and the integer restrictions on the remaining \mathbf{y} 's are relaxed, the resulting continuous subproblem (in the \mathbf{x} and relaxed \mathbf{y} variables) is convex. Although many practical problems meet these conditions, unfortunately many do not, often because some of the equality constraint functions $\mathbf{h}(\mathbf{x})$ are nonlinear. Then you cannot guarantee that the feasible region of each relaxed subproblem is convex, so local solutions may exist that are not global solutions. Consequently, a local NLP solver may terminate at a local solution that is not global in some tree node, and, in a minimization problem, the objective function value (call it “local”) is larger than the true optimal value. When the “local” value is tested to see if it exceeds the current upper bound, it may pass this test, and the node will be classified as “fathomed.” No further branches are allowed from this node. The “fathomed” classification is false if the true global optimal value at the node is less

than the current upper bound. Thus, the BB procedure fails to find any better solutions reached by further branching from this node. A nonoptimal solution to the MINLP may result.

EXAMPLE 9.3 OPTIMAL SELECTION OF PROCESSES

This problem, taken from Floudas (1995), involves the manufacture of a chemical C in process 1 that uses raw material B (see Figure E9.3a). B can either be purchased or manufactured via two processes, 2 or 3, both of which use chemical A as a raw material. Data and specifications for this example problem, involving several nonlinear input–output relations (mass balances), are shown in Table E9.3A. We want to determine which processes to use and their production levels in order to maximize profit. The processes represent design alternatives that have not yet been built. Their fixed costs include amortized design and construction costs over their anticipated lifetime, which are incurred only if the process is used.

To model this problem as a MINLP problem, we first assign the continuous variables to the different streams to represent the flows of the different chemicals. A_2 and A_3 are the amounts of A consumed by processes 2 and 3, B_2 and B_3 are the amounts of B produced by these processes, BP is the amount of B purchased in an external market, and C_1 is the amount of C produced by this process. We also define the 0–1 variables, Y_1 , Y_2 , and Y_3 to represent the existence of each of the processes.

The constraints in this problem are

1. Conversion

$$C_1 = 0.9B_1$$

$$B_2 = \ln(1 + A_2)$$

$$B_3 = 1.2 \ln(1 + A_3) \tag{a}$$

2. Mass balance for B

$$B_1 = B_2 + B_3 + BP \tag{b}$$

The specifications and limits that apply are as follows:

3. Nonnegativity condition for continuous variables

$$A_2, A_3, B_1, B_2, B_3, BP, C_1 \geq 0 \tag{c}$$

4. Integer constraints

$$Y_1, Y_2, Y_3 = 0 \text{ or } 1 \tag{d}$$

5. Maximum demand for C

$$C_1 \leq 1 \tag{e}$$

6. Limits on plant capacity

$$B_2 \leq 4Y_2$$

$$B_3 \leq 5Y_3$$

$$C_1 \leq 2Y_1 \tag{f}$$

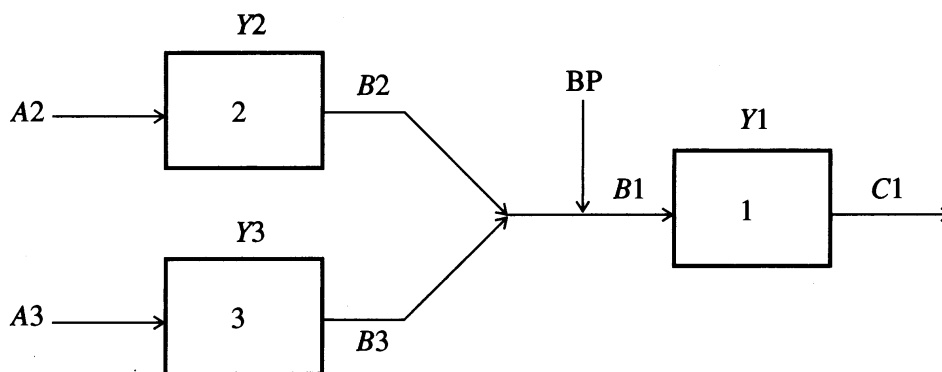


FIGURE E9.3a
Process diagram.

TABLE E9.3A
Problem data

Conversions:	Process	1	$C = 0.9B$
	Process	2	$B = \ln(1 + A)$
	Process	3	$B = 1.2 \ln(1 + A)$ (A, B, C , in ton/h)
Maximum capacity:	Process	1	2 ton/h of C
	Process	2	4 ton/h of B
	Process	3	5 ton/h of B
Prices:	A :	\$ 1,800/ton	
	B :	\$ 7,000/ton	
	C :	\$13,000/ton	
Demand of C :	1 ton/h maximum		
		Fixed	Variable
		(10^3 \$/h)	(10^3 \$/ton of product)
Costs:	Process 1	3.5	2
	Process 2	1	1
	Process 3	1.5	1.2

Note that the constraints in (f) place an upper limit of zero on the amounts produced if a process is not selected and impose the true upper limit if the process is selected. Clearly, with the bounds in step 3, this means that the amounts of B_2 , B_3 , and C_1 are zero when their binary variables are set to zero. If a binary variable is one, the amounts produced can be anywhere between zero and their upper limits.

Finally, for the objective function, the terms for the profit PR expressed in $\$10^3/h$ are given as follows:

1. Income from sales of product C : $13C$
2. Expense for the purchase of chemical B : $7BP$

3. Expense for the purchase of chemical A: $1.8A2 + 1.8A3$
4. Annualized investment or fixed cost for the three processes:

$$3.5Y1 + 2C1 + Y2 + B2 + 1.5Y3 + 1.2B3$$

Note that in the preceding expression the fixed charges are multiplied by the binary variables so that these charges are incurred only if the corresponding process is selected. Combining the preceding terms yields the following objective function:

$$\begin{aligned} \text{Maximize } PR = & 11C1 - 3.5Y1 - Y2 - B2 - 1.5Y3 - 1.2B3 \\ & - 7BP - 1.8A2 - 1.8A3 \end{aligned} \quad (g)$$

Relations (a)–(g) define the MINLP problem. It is important to note that the relations between the binary and continuous variables in Equation (f) are *linear*. It is possible to impose the desired relations nonlinearly. For example, one could replace $C1$ by $C1 * Y1$ everywhere $C1$ appears. Then if $Y1 = 0$, $C1$ does not appear, and if $Y1 = 1$, $C1$ does appear. Alternatively, one could replace $C1$ by the conditional expression (if $Y1 = 1$ then $C1$ else 0). Both these alternatives create nonlinear models that are very difficult to solve and should be avoided if possible.

Solution. Figure E9.3b shows the implementation of the MINLP problem in Excel Solver. The input–output relations (a) are in cells F18:F20, and the mass balance (b) is in F22, both written in the form “ $f(\mathbf{x}) = 0$.” The left- and right-hand sides of the plant capacity limits (f) are in C25:C27 and F25:F27, respectively. The Solver parameter dialog box is in Figure E9.3c. Nonnegativity constraints are imposed by checking the “Assume Nonnegative” box in the options dialog box.

The optimal solution has $Y1 = Y3 = 1$, $Y2 = 0$, so only processes 1 and 3 are used. Because $BP = 0$, there is no purchase of chemical B from an outside source. Total costs are 11.077 (in thousands of dollars per hour), revenues are 13, and the maximum profit is 1.923.

Given the optimal result, we now can ask a number of questions about the process operations, such as

1. Why is process 3 used instead of 2?
2. What happens if the cost of chemical A changes?
3. Why is no B purchased?

These questions can be answered, respectively, by carrying out the following steps:

1. Rerun the base case with $Y2$ fixed at 1 and $Y3$ at 0, thus forcing process 2 to be used rather than 3 while optimizing over the continuous flow variables.
2. Change the cost of A, and reoptimize.
3. Change the cost of purchased B, and reoptimize.

The link between the Excel Solver and the Excel Scenario Manager makes saving and reporting case study information easier. After solving each case, click the “Save Scenario” button on the dialog box that contains the optimality message, which invokes the Excel Scenario Manager. This stores the current decision variable values in a scenario named by the user. After all of the desired scenarios are generated, you can produce the Scenario Summary shown in Table E9.3B by selecting “Scenario Manager” from the Tools menu and choosing “Summary” from the Scenario Manager dialog.

	A	B	C	D	E	F
1	PROCESS MODEL SOLVED BY THE EXCEL SOLVER (BRANCH AND BOUND)					
2						
3	CONTINUOUS VARIABLES					
4	1. a2=consumption of chemical A in process 2					0.00E+00
5	2. a3=consumption of chemical A in process 3					1.52
6	3. b2=production of chemical B in process 2					0.00E+00
7	4. b3=production of chemical B in process 3					1.111
8	5. bp=amount of B purchased in external market					0.00E+00
9	6. b1=consumption of B by process 1					1.111
10	7. c1=amount of C produced by process 1					1
11	BINARY VARIABLES					
12	1. y1=on/off for process 1					1
13	2. y2=on/off for process 2					0
14	3. y3=on/off for process 3					1
15	CONSTRAINTS					
16	1. PROCESS OUTPUTS					
17						
18	inout1					0.00E+00
19	inout2					0.00E+00
20	inout3					0.00E+00
21	2. MASS BALANCE					
22	mbal					0.00E+00
23	3. TURN PROCESS OUTPUTS ON OR OFF					
24			Output	Capacity	Binary	Capacity*Binary
25	process 1		1	2	1	2
26	process 2		0.00E+00	4	0	0
27	process 3		1.111	5	1	5
28	4. OUTPUT LIMIT					
29			Value of c1	Limit		
30	limit on c1		1	1		
31	OBJECTIVE		Per unit	Total		
32	revenue		13	13		
33	<u>fixed cost</u>					
34	y1		3.5	3.5		
35	y2		1	0		
36	y3		1.5	1.5		
37	subtotal			5		
38	<u>operating cost</u>					
39	c1		2	2		
40	b2		1	0		
41	b3		1.2	1.333		
42	subtotal			3.333		
43	<u>raw materials costs</u>					
44	a2 cost		1.8	0.000		
45	a3 cost		1.8	2.744		
46	bp cost		7	0.000		
47	subtotal			2.744		
48	total cost			11.077		
49	net profit			1.923		

FIGURE E9.3b

Excel Solver model. Permission by Microsoft.

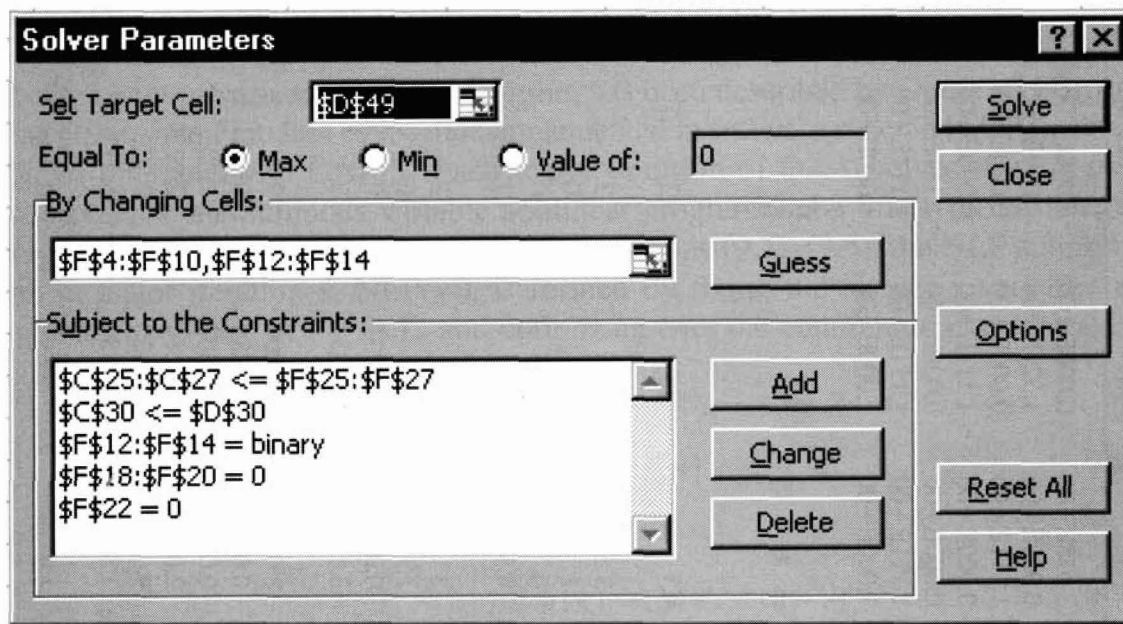


FIGURE E9.3c
Solver parameter dialog box. Permission by Microsoft.

Examination of the “2 instead of 3” column in Table E9.3B shows that process 3 has higher fixed and variable operating costs than 2 (3.33 compared with 3.11) but is more efficient because its output of B is 1.2 times that of process 2. This higher efficiency leads to lower raw material costs for chemical A (A_3 cost is 2.744, and A_2 cost is 3.668). This more than offsets the higher operating cost, leading to lower total costs and a larger net profit. This analysis clearly shows that the choice between processes 2 and 3 depends on the cost of A . If the cost of A is reduced enough, process 2 should be preferred. The two “ $acost$ ” columns in Table E9.3B show that a reduction of A ’s cost to 1.5 reduces the cost but leaves process 3 as the best choice, but a further reduction to 1.0 switches the optimal choice to process 2.

The last row of Table E9.3B shows why no chemical B is purchased. The cost per unit of B produced is computed by adding the cost of A purchased to the sum of the fixed and variable operating costs (processes 2 and 3) and dividing by the amount of B produced. In the base case this cost is \$2555/ton, so that the market price of B must be lower than this value for an optimal solution to choose purchasing B to producing it. The current price of B is 7, far above this threshold. The “ $BP_{cost}=2$ ” column of Table E9.3B shows that if B ’s market price is reduced to 2, the maximum profit is attained by shutting down both processes 2 and 3 and purchasing B .

Of course, a BB method can find an optimal solution even when the MINLP does not satisfy the convexity conditions. That occurred in Example 9.3, even though the equality constraints were nonlinear. The GRG2 solver did find global solutions at each node. An optimal solution cannot be guaranteed for nonconvex MINLPs, however, if a local NLP solver is used. As global optimization methods improve, future BB software may include a global NLP solver and thus ensure optimality. Currently, the main drawback to using a global optimizer in a BB algorithm is the long time required to find a global solution to even moderately-sized nonconvex NLPs.

Scenario summary

Changing cells	Base	2 instead of 3	<i>acost</i> =1	<i>acost</i> =1.5	<i>BPcost</i> =2
1. <i>A2</i> =consumption of chemical <i>A</i> in process 2	0.000E+00	2.038E+00	2.038E+00	0.000E+00	0.000E+00
2. <i>A3</i> =consumption of chemical <i>A</i> in process 3	1.524E+00	0.000E+00	0.000E+00	1.524E+00	0.000E+00
3. <i>B2</i> =production of chemical <i>B</i> by process 2	0.000E+00	1.111E+00	1.111E+00	0.000E+00	0.000E+00
4. <i>B3</i> =production of chemical <i>B</i> by process 3	1.111E+00	0.000E+00	0.000E+00	1.111E+00	0.000E+00
5. <i>BP</i> =amount of <i>B</i> purchased in external market	0.000E+00	0.000E+00	0.000E+00	0.000E+00	1.111E+00
6. <i>B1</i> =consumption of <i>B</i> by process 1	1.111E+00	1.111E+00	1.111E+00	1.111E+00	1.111E+00
7. <i>C1</i> =amount of <i>C</i> produced by process 1	1.000E+00	1.000E+00	1.000E+00	1.000E+00	1.000E+00
1. <i>Y1</i> = on-off for process 1	1	1	1	1	1
2. <i>Y2</i> = on-off for process 2	0	1	1	0	0
3. <i>Y3</i> = on-off for process 3	1	0	0	1	0
Result cells					
Revenue	13	13	13	13	13
Fixed cost					
<i>Y1</i>	3.5	3.5	3.5	3.5	3.5
<i>Y2</i>	0	1	1	0	0
<i>Y3</i>	1.5	0	0	1.5	0
Subtotal	5	4.5	4.5	5	3.5
Operating cost					
<i>C1</i>	2	2	2	2	2
<i>B2</i>	0	1.111	1.111	0	0
<i>B3</i>	1.333	0.000	0.000	1.333	0
Subtotal	3.333	3.111	3.111	3.333	2
Raw material costs					
<i>A2</i> cost	0.000	3.668	2.038	0.000	0
<i>A3</i> cost	2.744	0.000	0.000	2.286	0
<i>BP</i> cost	0.000	0.000	0.000	0.000	2.222
Subtotal	2.744	3.668	2.038	2.286	2.222
Total cost	11.077	11.279	9.649	10.620	7.722
Net profit	1.923	1.721	3.351	2.380	5.278
Unit cost of <i>B</i> produced	2.55	1.9	1.9	2.55	0

9.4 SOLVING MINLPs USING OUTER APPROXIMATION

The “outer approximation” (OA) algorithm has been described by Duran and Grossman (1986) and Floudas (1995). It is implemented in software called DICOPT, which has an interface with GAMS. Each major iteration of OA involves solving two subproblems: a continuous variable nonlinear program and a linear mixed-integer program. Using the problem statement in Equations (9.2)–(9.5), the NLP subproblem at major iteration k , $\text{NLP}(\mathbf{y}^k)$, is formed by fixing the integer \mathbf{y} variables at some set of values, say $\mathbf{y}^k \in Y$, and optimizing over the continuous \mathbf{x} variables;

Problem NLP (\mathbf{y}^k)

$$\text{Maximize: } \mathbf{c}^T \mathbf{y}^k + f(\mathbf{x}) \quad (9.6)$$

$$\text{Subject to: } \mathbf{h}(\mathbf{x}) = \mathbf{0}$$

$$\mathbf{g}(\mathbf{x}) + \mathbf{M}\mathbf{y}^k \leq \mathbf{0} \quad (9.7)$$

$$\mathbf{x} \in X$$

We redefined the sense of the optimization to be maximization. The optimal objective value of this problem is a lower bound on the MINLP optimal value. The MILP subproblem involves both the \mathbf{x} and \mathbf{y} variables. At iteration k , it is formed by linearizing all nonlinear functions about the optimal solutions of each of the subproblems $\text{NLP}(\mathbf{y}^i)$, $i = 1, \dots, k$, and keeping all of these linearizations. If \mathbf{x}^i solves $\text{NLP}(\mathbf{y}^i)$, the MILP subproblem at iteration k is

MILP subproblem

$$\text{Maximize: } \mathbf{c}^T \mathbf{y} + z \quad (9.8)$$

$$\text{Subject to: } z \geq f(\mathbf{x}^i) + \nabla f^T(\mathbf{x}^i)(\mathbf{x} - \mathbf{x}^i), \quad i = 1, \dots, k$$

$$\mathbf{h}(\mathbf{x}^i) + \nabla \mathbf{h}^T(\mathbf{x}^i)(\mathbf{x} - \mathbf{x}^i) = \mathbf{0}, \quad i = 1, \dots, k \quad (9.9)$$

$$\mathbf{g}(\mathbf{x}^i) + \nabla \mathbf{g}^T(\mathbf{x}^i)(\mathbf{x} - \mathbf{x}^i) + \mathbf{M}\mathbf{y}^i \leq \mathbf{0}, \quad i = 1, \dots, k$$

$$\mathbf{x} \in X, \quad \mathbf{y} \in Y$$

The new variable z is introduced to make the objective linear.

$$\text{Minimize: } \mathbf{c}^T \mathbf{y} + z$$

$$\text{Subject to: } z \geq f(\mathbf{x})$$

is equivalent to minimizing $\mathbf{c}^T \mathbf{y} + f(\mathbf{x})$. Duran and Grossman (1986) and Floudas (1995) show that if the convexity assumptions (1)–(5) of Section 9.3 hold, then the optimal value of this MILP subproblem is an upper bound on the optimal MINLP objective value. Because a new set of linear constraints is added at each iteration, this upper bound decreases (or remains the same) at each iteration. Under the convexity

TABLE 9.1
DICOPT iteration log

Major step	Major iteration	Objective function	CPU time (s)	NLP or MILP iterations	Solver
NLP	1	5.32542	0.17	8	CONOPT2
MILP	1	2.44260	0.16	16	OSL
NLP	2	1.72097	0.11	3	CONOPT2
MILP	2	2.20359	0.16	17	OSL
NLP	3	1.92310	0.11	3	CONOPT2
MILP	3	1.44666	0.17	24	OSL
NLP	4	1.41100	0.11	8	CONOPT2

assumptions, the upper and lower bounds converge to the true optimal MINLP value in a finite number of iterations, so the OA algorithm solves the MINLP problem.

Table 9.1 shows how outer approximation, as implemented in the DICOPT software, performs when applied to the process selection model in Example 9.3. Note that this model does not satisfy the convexity assumptions because its equality constraints are nonlinear. Still DICOPT does find the optimal solution at iteration 3. Note, however, that the optimal MILP objective value at iteration 3 is 1.446, which is *not* an upper bound on the optimal MINLP value of 1.923 because the convexity conditions are violated. Hence the normal termination condition that the difference between upper and lower bounds be less than some tolerance cannot be used, and DICOPT may fail to find an optimal solution. Computational experience on nonconvex problems has shown that retaining the best feasible solution found thus far, and stopping when the objective value of the NLP subproblem fails to improve, often leads to an optimal solution. DICOPT stopped in this example because the NLP solution at iteration 4 is worse (lower) than that at iteration 3.

The NLP solver used by GAMS in this example is CONOPT2, which implements a sparsity—exploiting GRG algorithm (see Section 8.7). The mixed-integer linear programming solver is IBM's Optimization Software Library (OSL). See Chapter 7 for a list of commercially available MILP solvers.

9.5 OTHER DECOMPOSITION APPROACHES FOR MINLP

Generalized Benders decomposition (GBD), derived in Geoffrion (1972), is an algorithm that operates in a similar way to outer approximation and can be applied to MINLP problems. Like OA, when GBD is applied to models of the form (9.2)–(9.5), each major iteration is composed of the solution of two subproblems. At major iteration k , one of these subproblems is $NLP(y^k)$, given in Equations (9.6)–(9.7). This is an NLP in the continuous variables x , with y fixed at y^k . The other GBD subproblem is an integer linear program, as in OA, but it only involves the

discrete variables \mathbf{y} , whereas the MILP of OA involves both \mathbf{x} and \mathbf{y} . The constraints of the GBD MIP subproblem are different from those in the OA MILP subproblem. These constraints are called generalized Benders cuts. They are linear constraints, formed using the Lagrange multipliers of the continuous subproblem, $\text{NLP}(\mathbf{y}^k)$. Only one GBD cut is added at each major iteration. In OA, an entire set of linearized constraints of the form (9.9) is added each time, so the OA MILP subproblems has many more constraints than those in GBD. Each solution of the NLP subproblem in GBD generates a lower bound on the maximum objective value, and the MILP subproblem yields an upper bound. Duran and Grossman (1986) proved that for convex MINLP problems the OA upper bound is never above the GBD lower bound [see also Floudas (1995)]. Hence, for convex problems OA terminates in fewer major iterations than GBD. The OA computing time may not be smaller than that for GBD, however, because the OA subproblems have more constraints and thus usually take longer to solve.

9.6 DISJUNCTIVE PROGRAMMING

A disjunctive program is a special type of MINLP problem whose constraints include the condition that exactly one of several sets of constraints must be satisfied (Raman and Grossmann, 1994). Defining \vee as the logical “exclusive or” operator and Y_i as logical variables (whose values are *true* or *false*), an example of a disjunctive program, taken from Lee and Grossman (2000), is

$$\begin{aligned} &\text{Minimize: } (x_1 - 3)^2 + (x_2 - 2)^2 + c \\ &\text{Subject to: } \left[\begin{array}{c} Y_1 \\ x_1^2 + x_2^2 - 1 \leq 0 \\ c = 2 \end{array} \right] \vee \left[\begin{array}{c} Y_2 \\ (x_1 - 4)^2 + (x_2 - 1)^2 - 1 \leq 0 \\ c = 1 \end{array} \right] \\ &\qquad\qquad\qquad \vee \left[\begin{array}{c} Y_3 \\ (x_1 - 2)^2 + (x_2 - 4)^2 - 1 \leq 0 \\ c = 3 \end{array} \right] \end{aligned}$$

and

$$0 \leq x_i \leq 8, \quad i = 1, 2$$

The logical condition, called a *disjunction*, means that exactly one of the three sets of conditions in brackets must be true: the logical variable must be true, the constraint must be satisfied, and c must have the specified value. Note that c appears in the objective function. There are additional constraints on \mathbf{x} ; here these are simple bounds, but in general they can be linear or nonlinear inequalities. The single inequality constraint in each bracket may be replaced by several different inequalities. There

may also be logical constraints on the Y_i variables, but these constraints are not included in this example.

Disjunctions arise when a set of alternative process units is considered during a process design. The following example is taken from Biegler et al. (1997), p. 519. If one of two reactors is to be selected, we may have the conditions:

If reactor one is selected, then pressure P in the reactor must lie between 10 and 15, and the reactor fixed cost c is 20.

If reactor two is selected, then pressure P in the reactor must be between 5 and 10 and the reactor fixed cost c is 30.

See Hooker and Grossman (1999) for more details on occurrence of disjunctions in process synthesis problems.

A generalized disjunctive program (GDP) may be formulated as an MINLP, with binary variables y_i replacing the logical variables Y_i . The most common formulation is called the “big- M ” approach because it uses a large positive constant denoted by M to relax or enforce the constraints. This formulation of the preceding example follows:

$$\begin{aligned} \text{Minimize: } & (x_1 - 3)^2 + (x_2 - 2)^2 + 2y_1 + y_2 + 3y_3 \\ \text{Subject to: } & x_1^2 + x_2^2 - 1 \leq M(1 - y_1) \\ & (x_1 - 4)^2 + (x_2 - 1)^2 - 1 \leq M(1 - y_2) \\ & (x_1 - 2)^2 + (x_2 - 4)^2 - 1 \leq M(1 - y_3) \\ & y_1 + y_2 + y_3 = 1 \end{aligned}$$

and

$$y_i = 0 \quad \text{or} \quad 1, \quad i = 1, 2, 3, \quad 0 \leq x_i \leq 8, \quad i = 1, 2$$

When $y_i = 1$, the i th constraint is enforced and the correct value of c is added to the objective. When $y_i = 0$, the right-hand side of the i th constraint is equal to M , so it is never active if M is large enough. The constraint that sets the sum of the y_i equal to 1 ensures that exactly one constraint is enforced.

The big- M formulation is often difficult to solve, and its difficulty increases as M increases. This is because the NLP relaxation of this problem (the problem in which the condition $y_i = 0$ or 1 is replaced by y_i between 0 and 1) is often weak, that is, its optimal objective value is often much less than the optimal value of the MINLP. An alternative to the big- M formulation is described in Lee and Grossman (2000) using an NLP relaxation, which often has a much tighter bound on the optimal MINLP value. A branch-and-bound algorithm based on this formulation performed much better than a similar method applied to the big- M formulation. An outer approximation approach is also described by Lee and Grossmann (2000).

REFERENCES

Biegler, L. T.; I. E. Grossmann; and A. W. Westerberg. *Systematic Methods of Chemical Process Design*. Prentice-Hall, Englewood Cliffs, NJ (1997).

- Duran, M. A.; and I. E. Grossmann. "An Outer Approximation Algorithm for a Class of Mixed—Integer Nonlinear Programs." *Math Prog* **36**: 307–339, (1986).
- Floudas, C. *Nonlinear and Mixed—Integer Optimization*. Oxford University Press, New York (1995).
- Fylstra, D.; L. Lasdon; A. Waren, and J. Watson. "Design and Use of the Microsoft Excel Solver." *Interfaces* **28** (5): 29–55, (Sept–Oct, 1998).
- Geoffrion, A. M. "Generalized Benders Decomposition." *J Optim Theory Appl* **10**(4): 237–260 (1972).
- Grossmann, I. E.; and J. Hooker. *Logic Based Approaches for Mixed Integer Programming Models and Their Application in Process Synthesis*. FOCAPD Proceedings, AIChE Symp. Ser. **96** (323): 70–83 (2000).
- Hillier, F. S.; and G. J. Lieberman. *Introduction to Operations Research*, 4th ed. Holden-Day, San Francisco, CA (1986), p. 582.
- Lee, S.; and I. E. Grossmann. "New Algorithms for Nonlinear Generalized Disjunctive Programming." In press, *Comput Chem Engr*.
- Murray, J. E.; and T. F. Edgar. "Optimal Scheduling of Production and Compression in Gas Fields." *J Petrol Technol* 109–118 (January, 1978).
- Nemhauser, G. L.; and L. A. Wolsey. *Integer and Combinatorial Optimization*. J. Wiley, New York (1988).
- Raman, R.; and I. E. Grossmann. "Modeling and Computational Techniques for Logic Based Integer Programming." *Comput Chem Engr* **18**: 563–578 (1994).
- Rosenwald, G. W.; and D. W. Green. "A Method for Determining the Optimal Location of Wells." *Soc Petrol Eng J* pp. 44–54 (February, 1974).

SUPPLEMENTARY REFERENCES

- Adjiman, C. S.; I. P. Androulakis; and C. A. Floudas. "Global Optimization of MINLP Problems in Process Synthesis and Design." *Comput Chem Eng* **21**: S445–450 (1997).
- Crowder, H. P.; E. L. Johnson; and M. W. Padberg. "Solving Large-Scale Zero-One Linear Programming Problems." *Oper Res* **31**: 803–834 (1983).
- Galli, M. R.; and J. Cerda. "A Customized MILP Approach to the Synthesis of Heat Recovery Networks Reaching Specified Topology Targets." *Ind Eng Chem Res* **37**: 2479–2486 (1998).
- Grossmann, I. E. "Mixed-Integer Optimization Techniques for Algorithmic Process Synthesis." *Adv Chem Eng* **23**: 172–239, (1996).
- Grossmann, I. E.; J. A. Caballero; and H. Yeomans. "Mathematical Programming Approaches to the Synthesis of Chemical Process Systems." *Korean J Chem Eng* **16**(4): 407–426 (1999).
- Grossmann, I. E.; and Z. Kravanja. "Mixed-Integer Nonlinear Programming: A Survey of Algorithms and Applications." In *Large-Scale Optimization with Applications, Part 2: Optimal Design and Control*, L. T. Biegler et al., eds., pp. 73–100, Springer-Verlag, New York (1997).
- Mokashi, S. D.; and A. Kokossis. "The Maximum Order Tree Method: A New Approach for the Optimal Scheduling of Product Distribution Lines." *Comput Chem Eng* **21**: S679–684 (1997).
- Morari, M.; and I. Grossmann (eds.). *CACHE Process Design Case Studies, Volume 6: Chemical Engineering Optimization Models with GAMS* (October, 1991). CACHE Corporation, Austin, TX.

- Parker, R. G.; and R. Rardin. *Discrete Optimization*. Academic Press, New York (1986).
- Schrijver, A. *Theory of Linear and Integer Programming*. Wiley-Interscience, New York (1986).
- Skrifvars, H.; S. Leyffer; and T. Westerlund. "Comparison of Certain MINLP Algorithms When Applied to a Model Structure Determination and Parameter Estimation Problem." *Comput Chem Eng* **22**: 1829–1835 (1998).
- Turkay, M.; and I. E. Grossmann. "Logic-Based MINLP Algorithms for the Optimal Synthesis of Process Networks." *Comput Chem Eng* **20** (8): 959–978 (1996).
- Westerlund, T.; H. Skrifvars; I. Harjunkoski, and R. Pom. "An Extended Cutting Plane Method for a Class of Non-convex MINLP Problems." *Comput Chem Eng* **22**: 357–365 (1998).
- Xia, Q.; and S. Macchietto. "Design and Synthesis of Batch Plants—MINLP Solution Based on a Stochastic Method." *Comput Chem Eng* **21**: S697–702 (1997).
- Yi, G.; Suh, K.; Lee, B.; and E. S. Lee. "Optimal Operation of Quality Controlled Product Storage." *Comput Chem Eng* **24**: 475–480 (2000).
- Zamora, J. M.; and I. E. Grossmann. "A Global MINLP Optimization Algorithm for the Synthesis of Heat Exchanger Networks with No Stream Splits." *Comput Chem Eng* **22**: 367–384 (1998).

PROBLEMS

9.1 A microelectronics manufacturing facility is considering six projects to improve operations as well as profitability. Due to expenditure limitations and engineering staffing constraints, however, not all of these projects can be implemented. The following table gives projected cost, staffing, and profitability data for each project.

Project	Description	First-year expenditure (\$)	Second-year expenditure (\$)	Engineering hours	Net present value (\$)
1	Modify existing production line with new etchers	300,000	0	4000	100,000
2	Build new production line	100,000	300,000	7000	150,000
3	Automate new production line	0	200,000	2000	35,000
4	Install plating line	50,000	100,000	6000	75,000
5	Build waste recovery plant	50,000	300,000	3000	125,000
6	Subcontract waste disposal	100,000	200,000	600	60,000

The resource limitations are

First-year expenditure:	\$450,000
Second-year expenditure:	\$400,000
Engineering hours:	10,000

A new or modernized production line must be provided (project 1 or 2). Automation is feasible only for the new line. Either project 5 or project 6 can be selected, but not both. Determine which projects maximize the net present value subject to the various constraints.

- 9.2** An electric utility must determine which generators to start up at the beginning of each day. They have three generators with capacities, operating cost, and start-up costs shown in the following table. A day is divided into two periods, and each generator may be started at the beginning of each period. A generator started in period 1 may be used in period 2 without incurring an additional start-up cost. All generators are turned off at the end of the day.

Demand for power is 2500 megawatts (MW) in period 1 and 3500 MW in period 2. Formulate and solve this problem as a mixed-integer linear program. Define the binary variables carefully.

Generator	Fixed start-up cost (\$)	Cost per period per megawatt	Generator capacity in each period (MW)
1	2800	5	1900
2	2000	3	1700
3	1900	8	2900

- 9.3** An electric utility currently has 700 MW of generating capacity and needs to expand this capacity over the next 5 years based on the following demand forecasts, which determine the minimum capacity required.

Year	Minimum capacity (MW)
1	780
2	860
3	950
4	1060
5	1180

Capacity is increased by installing 10-, 50-, or 100-MW generators. The cost of installation depends on the size and year of installation as shown in the following table.

Generator size (MW)	Year 1	Year 2	Year 3	Year 4	Year 5
10	280	230	188	153	135
50	650	538	445	367	300
100	700	771	640	530	430

Once a generator is installed, it is available for all future years. Formulate and solve the problem of determining the amount of new capacity to install each year so that minimum capacities are met or exceeded and total (undiscounted) installation cost is minimized.

- 9.4** A manufacturing line makes two products. Production and demand data are shown in the following table.

	Product 1	Product 2
Set-up time (hrs)	6	11
Set-up cost (\$)	250	400
Production time/unit (h)	0.5	0.75
Production cost/unit (\$)	9	14
Inventory holding cost/unit	3	3
Penalty cost for unsatisfied demand/unit (\$)	15	20
Selling price (\$/unit)	25	35

Demand data				
Product	Week 1	Week 2	Week 3	Week 4
1	75	95	60	90
2	20	30	45	30

Total time available (for production and setup) in each week is 80 h. Starting inventory is zero, and inventory at the end of week 4 must be zero. Only one product can be produced in any week, and the line must be shut down and cleaned at the end of each week. Hence the set-up time and cost are incurred for a product in any week in which that product is made. No production can take place while the line is being set up.

Formulate and solve this problem as an MILP, maximizing total net profit over all products and periods.

- 9.5** A portfolio manager has \$100,000 to invest in a list of 20 stocks. She estimates the return from stock i over the next year as $r(i)$, so that if $x(i)$ dollars are invested in stock i at the start of the year, the end of year value is $[1 + r(i)] * x(i)$. Write an MILP model that determines the amounts to invest in each stock in order to maximize end-of-year portfolio value under the following investment policy: no more than \$20,000 can be invested in any stock, and if a stock is purchased at all, at least \$5000 worth must be purchased.

9.6

$$\text{Maximize: } f(\mathbf{x}) = 75x_1 + 6x_2 + 3x_3 + 33x_4$$

$$\text{Subject to: } 774x_1 + 76x_2 + 22x_3 + 42x_4 \leq 875$$

$$67x_1 + 27x_2 + 794x_3 + 53x_4 \leq 875$$

$$x_1, x_2, x_3, x_4 \text{ either 0 or 1}$$

9.7

$$\text{Maximize: } f(\mathbf{x}) = 2x_1 + x_2$$

$$\text{Subject to: } x_1 + x_2 \leq 5$$

$$x_1 - x_2 \geq 0$$

$$6x_1 + 2x_2 \leq 21$$

$$x_1, x_2 \geq 0 \text{ and integer}$$

9.8

$$\text{Minimize: } f(\mathbf{x}) = x_1 + 4x_2 + 2x_3 + 3x_4$$

$$\text{Subject to: } -x_1 + 3x_2 - x_3 + 2x_4 \geq 2$$

$$x_1 + 3x_2 + x_3 + x_4 \geq 3$$

$$x_1, x_2 \geq 0 \text{ and integer}$$

$$x_3, x_4 \geq 0$$

9.9 Determine the minimum sum of transportation costs and fixed costs associated with two plants and two customers based on the following data:

Plant	Annual capacity (in thousands)	Annual fixed charges (in 10^4)
1	2	1
2	1	1

Customer (j)	Demand (j)
1	1
2	1

Plant (i)	Customer (j)	
	1	2
1	$\frac{1}{3}$	$\frac{1}{2}$
2	0	1

Hint: The mathematical statement is

$$\text{Minimize: } f(\mathbf{x}) = \sum_i \sum_j C_{ij}^T x_{ij} + \sum_i C_i^F y_i$$

$$\text{Subject to: } \sum_i x_{ij} = D_j, \quad j = 1, \dots, n$$

$$\sum_j x_{ij} \leq A_i, \quad i = 1, \dots, m$$

$$x_{ij} - \min \{D_j, A_j\} \leq 0, \quad i = 1, \dots, m \quad j = 1, \dots, n$$

$$y_i = 0, 1 \text{ (integers)}, \quad i = 1, \dots, m$$

where C_i^T = unit transportation cost from plant i to customer j

C_i^F = fixed cost associated with plant i

x_{ij} = quantity supplied to customer j from plant i

$y_i = 1$ (plant operates); $= 0$ (plant is closed)

A_i = capacity of plant i

D_j = demand of customer j

9.10 Four streams are to be allocated to four extractors. The costs of each stream are

Stream	Extractor			
	1	2	3	4
1	45	P	5	56
2	27	2	82	74
3	19	55	3	P
4	3	10	4	84

The symbol P means the transfer is prohibited. Minimize the total costs.

9.11

$$\text{Minimize: } f(\mathbf{x}) = 10x_1 + 11x_2$$

$$\text{Subject to: } 9x_1 + 11x_2 \geq 29$$

$$\mathbf{x} \geq 0 \text{ and integer}$$

9.12

$$\text{Maximize: } f = 5x_1 + 8x_2 + 6x_3$$

$$\text{Subject to: } 9x_1 + 6x_2 + 10x_3 \leq 14$$

$$20x_1 + 63x_2 + 10x_3 \leq 110$$

$$x_i \geq 0, \text{ integer}$$

9.13

$$\text{Maximize: } f = x_1 + x_2 + x_3$$

$$\text{Subject to: } x_1 + 2x_2 + 2x_3 + 2x_4 + 3x_5 \leq 18$$

$$2x_1 + x_2 + 2x_3 + 3x_4 + 2x_5 \leq 15$$

$$x_1 \qquad \qquad \qquad -6x_4 \qquad \leq 0$$

$$\qquad \qquad x_2 \qquad \qquad \qquad -8x_5 \leq 0$$

$$\text{all } x_j \geq 0, \text{ integer}$$

9.14 A plant location problem has arisen. Two possible sites exist for building a new plant, A and B , and two customer locations are to be supplied, C and D . Demands and production/supply costs are listed as follows.

Use the following notation to formulate the optimization problem, and solve it for the values of I_1 and I_2 as well as the values of S_{ij} . Each plant has a maximum capacity of 500 units per day.

I_i = decision variable (0–1) associated with the decision to build, or not to build, a plant in a given location, and thus incurs the associated fixed daily cost.

C_{ij} = unit cost of supplying customer j from plant i .

C_i = fixed daily cost of plant i

S_{ij} = quantity supplied from the i th plant to the j th customer

R_j = requirement of j th customer

Q_i = capacity of proposed plant

Production and transport costs per unit:

A to C \$1.00

A to D \$3.00

B to C \$4.50

B to D \$1.00

Fixed plant charges per day:

plant A \$700

plant B \$610

Minimum demand (units per day):

Customer C 200

Customer D 250

9.15 The ABC company runs two refineries supplying three markets, using a pipeline owned by the XYZ company. The basic charge for pipeline use is \$80 per 1000 barrels. If more than 500 barrels are shipped from the refineries to one market, then the charge drops to \$60 per 1000 barrels for the next 1500 barrels. If more than 2000 barrels are shipped from the refineries to market, then the subsequent charge is \$40 per 1000 barrels for any over the 2000.

The objective is to meet demands at M_1 , M_2 , and M_3 , using supplies from R_1 and R_2 .

X_{ijk} = number of barrels from source i to destination j at price k .

C_{ijk} = shipping cost of X_{ijk}

I_{jk} = 0–1 variable to indicate whether or not any product is delivered to destination j at price level k .

We can state the general problem briefly as follows:

$$\text{Minimize: } \sum_{ijk} C_{ijk} x_{ijk} \quad (1)$$

$$\text{Subject to: } \sum_{ik} x_{ijk} \geq M_j \text{ for all } j \text{ (must meet demands)} \quad (2)$$

$$\text{and: } \sum_{jk} x_{ijk} \leq R_i \text{ for all } i \text{ (= sources) (cannot exceed supply)} \quad (3)$$

$$\sum_i x_{ij2} - b_{j2} I_{j2} \leq 0 \text{ for all } j \quad (4)$$

$$\sum_i x_{ij1} - b_{j1} I_{j2} \geq 0 \text{ for all } j \text{ (if any taken at second price must first use all at top price)} \quad (5)$$

$$\sum_i x_{ij3} - b_{j3} I_{j3} \leq 0 \text{ for all } j \quad (6)$$

$$\sum_i x_{ij2} - b_{j2} I_{j3} \geq 0 \text{ for all } j \text{ (if any taken at third price must first use all at second price)} \quad (7)$$

b_{jk} = upper bound on product delivered to terminal j at k th price level.

$$I_{j2} \leq 1 \text{ for all } j \text{ (upper bounds on integer variables)} \quad (8)$$

$$I_{j3} \leq 1 \text{ for all } j$$

The detailed matrix for this problem is set out in Table P9.15. Solve for the I_{jk} values and the x_{ijk} values.

TABLE P9.15 Variables

	x111	x112	x113	x121	x122	x123	x131	x132	x133	x211	x212	x213	x221	x222	x223	x231	x232	x233	112	113	122	123	132	133	
Upper bound	0.5	1.5		0.5	1.5		0.5	1.5		0.5	1.5		0.5	1.5		0.5	1.5		1	1	1	1	1	1	
Lower bound																			0	0	0	0	0	0	
Objective	80	60	40	80	60	40	80	60	40	80	60	40	80	60	40	80	60	40							
DEM.M1	1	1	1							1	1	1						40						≥ 2.0	
DEM.M2				1	1	1							1	1	1										≥ 5.0
DEM.M3							1	1	1							1	1	1							≥ 4.0
CAP.R1	1	1	1	1	1	1	1	1	1																≤ 5.0
CAP.R2										1	1	1	1	1	1	1	1	1							≤ 7.5
M1MAXP1	1									1									-0.5						≥ 0
M1MINP2		-1									-1								1.5						≥ 0
M1MAXP2		1									1									-1.5					≥ 0
M1MAXP3			-1									-1								0					≥ 0
M2MAXP1				1									1								-0.5				≥ 0
M2MINP2					-1									-1							1.5				≥ 0
M2MAXP2					1									1								-1.5			≥ 0
M2MAXP3						-1									-1							3.0			≥ 0
M3MAXP1							1									1							-0.5		≥ 0
M3MINP2								-1									-1						1.5		≥ 0
M3MAXP2								1									1							-1.5	≥ 0
M3MAXP3									-1										-1					2.0	≥ 0

GLOBAL OPTIMIZATION
FOR PROBLEMS WITH CONTINUOUS
AND DISCRETE VARIABLES

10.1	Methods for Global Optimization	382
10.2	Smoothing Optimization Problems	384
10.3	Branch-and-Bound Methods	385
10.4	Multistart Methods	388
10.5	Heuristic Search Methods	389
10.6	Other Software for Global Optimization	411
	References	412
	Supplementary References	413

IN CHAPTERS 6 AND 8 we showed that in continuous variable minimization problems with convex feasible regions and convex objectives, any local minimum is the global minimum. As discussed in Section 8.2, many problems do not satisfy these convexity conditions, and it is often difficult to verify whether they satisfy them or not. Models that include nonlinear equality constraints fall in this latter category. These constraints arise from nonlinear material balances (for which both flows and concentrations are unknowns), nonlinear physical property relations, nonlinear blending equations, nonlinear process models, and so on. Another source of nonconvexity can be in the objective function if it is concave, which can occur when production costs increase with the amount produced, but at a decreasing rate due to economies of scale. A problem involved in minimizing a concave objective function over a convex region defined by linear constraints is that it may have many local minima, one at each extreme point of the region. Nonconvex objective functions and local minima can also occur when you estimate the values of the model parameters using the least-squares or maximum likelihood objective functions.

Any problem containing discretely valued variables is nonconvex, and such problems may also be solved by the methods described in this chapter. The search methods discussed in Section 10.5 are often applied to supply chain and production-sequencing problems.

10.1 METHODS FOR GLOBAL OPTIMIZATION

If an NLP algorithm such as SLP, SQP, or GRG described in Chapter 8 is applied to a smooth nonconvex problem, it usually converges to the “nearest” local minimum, which may not be the global minimum. We refer to such algorithms in this chapter as “local solvers.” The problem of finding a global minimum is much more difficult than that of finding a local one, but several well-established general-purpose approaches to the problem are discussed subsequently.

Figure 10.1 shows a classification of global optimization methods. Exact methods, if allowed to run until they meet their termination criteria, are guaranteed to find an arbitrarily close approximation to a global optimum and to verify that they have done so. These include branch-and-bound (BB) methods, which were discussed in the context of mixed-integer linear and nonlinear programming in Chapter 9, methods based on interval arithmetic (Kearfott, 1996), and some multistart procedures, which invoke a local solver from multiple starting points. Heuristic search methods may and often do find global optimal solutions, but they are not guaranteed to do so, and we are usually unable to prove that they have found a global solution even when they have done so. Nonetheless, they are widely used, often find very good solutions, and can be applied to both mixed-integer and combinatorial problems. A heuristic search method starts with some current solution, explores all solutions in some neighborhood of that point looking for a better one, and repeats if an improved point is found. Metaheuristics algorithms guide and improve on a heuristic algorithm. These include tabu search, scatter search, simulated annealing, and genetic algorithms. They use a heuristic procedure for the

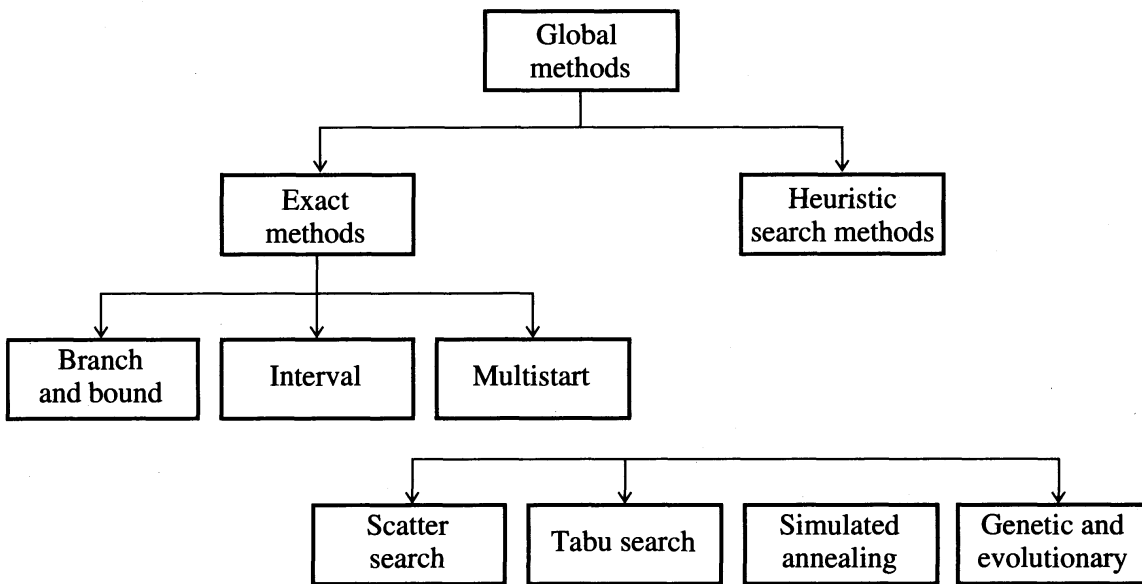


FIGURE 10.1
Classification of global optimization methods.

problem class, which by itself may not be able to find a global optimum, and guide the procedure by changing its logic-based search so that the method does not become trapped in a local optimum. Genetic and evolutionary algorithms use heuristics that mimic the biological processes of crossover and mutation. They are “population-based” methods that combine a set of solutions (the “population”) in an effort to find improved solutions and then update the population when a better solution is found. Scatter search is also a population-based procedure.

The methods mentioned earlier are general-purpose procedures, applicable to almost any problem. Many specialized global optimization procedures exist for specific classes of nonconvex problems. See Pinter (1996a) for a brief review and further references. Typical problems are

- Problems with concave objective functions to be minimized over a convex set.
- “Differential convex” (DC) problems of the form

$$\begin{aligned} \text{Minimize: } & f(\mathbf{x}) \\ \text{Subject to: } & g_j(\mathbf{x}) \leq \mathbf{0}, \quad j = 1, 2, \dots, J \\ & \mathbf{x} \in C \end{aligned}$$

and

where C is a convex set, and f and each constraint function g_j can be expressed as the difference of two convex functions, such as $f(x) = p(x) - q(x)$.

- Indefinite quadratic programs, in which the constraints are linear and the objective function is a quadratic function that is neither convex nor concave because its Hessian matrix is indefinite.
- Fractional programming problems, where the objective is a ratio of two functions.

If a problem has one of these forms, the special-purpose solution methods designed for it often produce better results than a general-purpose approach. In this

chapter, we focus on general-purpose methods and on frameworks that are applicable to wide classes of problems and do not discuss special problem classes further.

10.2 SMOOTHING OPTIMIZATION PROBLEMS

All gradient-based NLP solvers, including those described in Chapter 8, are designed for use on problems in which the objective and constraint functions have continuous first partial derivatives everywhere. Examples of functions that do not have continuous first partials everywhere are

1. $|f(x)|$
2. $\max(f(x), g(x))$
3. $h(x) = \{\text{if } x_1 \leq 0 \text{ then } f(x) \text{ else } g(x)\}$
4. A piecewise linear function interpolating a given set of (y_i, x_i) values.

If you encounter these functions, you can reformulate them as equivalent smooth functions by introducing additional constraints and variables. For example, consider the problem of fitting a model to n data points by minimizing the sum of weighted absolute errors between the measured and model outputs. This can be formulated as follows:

$$\text{Minimize: } e(\mathbf{x}) = \sum_{i=1}^n w_i |y_i - h(v_i, \mathbf{x})|$$

where \mathbf{x} = a vector of model parameter values

w_i = a positive weight for the error at the i th data point

y_i = the measured output of the system being modeled when the vector of system inputs is v_i

$h(v_i, \mathbf{x})$ = the calculated model output when the system inputs are v_i

This weighted sum of absolute values in $e(\mathbf{x})$ was also discussed in Section 8.4 as a way of measuring constraint violations in an exact penalty function. We proceed as we did in that section, eliminating the nonsmooth absolute value function by introducing positive and negative deviation variables dp_i and dn_i and converting this nonsmooth unconstrained problem into an equivalent smooth constrained problem, which is

$$\text{Minimize: } \sum_{i=1}^n w_i (dp_i + dn_i) \quad (10.1)$$

$$\text{Subject to: } y_i - h(v_i, \mathbf{x}) = dp_i - dn_i \quad i = 1, \dots, n \quad (10.2)$$

$$\text{and} \quad dp_i \geq 0, \quad dn_i \geq 0 \quad i = 1, \dots, n \quad (10.3)$$

In this problem, if the error is positive, then dp_i is positive and dn_i is zero in any optimal solution. For negative errors, dp_i is zero and dn_i is positive. The absolute

error is thus the sum of these deviation variables. A similar reformulation allows the problem of minimizing the maximum error to be posed as a smooth constrained problem.

If it is difficult or impossible to eliminate the nonsmooth functions by these or some other transformations, you can apply a gradient-based optimizer and hope that a nonsmooth point is never encountered. If one is encountered, the algorithm may fail to make further progress because the computed derivatives at the point are not meaningful. A large body of literature on methods for nonsmooth optimization exists (see Hiriart-Urruty and LeMarechal, 1993, for example), but software for nonsmooth optimization is not yet widely available. Alternatively, you can apply an optimization method that does not require first partial derivatives. Such algorithms include the Nelder–Meade simplex method (not the same as the simplex method for linear programming), the Hooke–Jeeves procedure, or a conjugate directions method due to Powell that does not use derivatives (Avriel, 1976). These techniques are not as sensitive to derivative discontinuities as gradient-based algorithms, but continually improve the objective function until they reach an approximation to a local minimum. They are not guaranteed to converge to a local solution for nonsmooth problems, and are basically unconstrained methods. You can incorporate constraints by using penalty functions; but if a large penalty weight is used, the objective function becomes ill-conditioned and hard to optimize with high accuracy. The search methods described in Section 10.5 are not as sensitive to discontinuities and are much less likely than a local solver to be trapped near a local optimum.

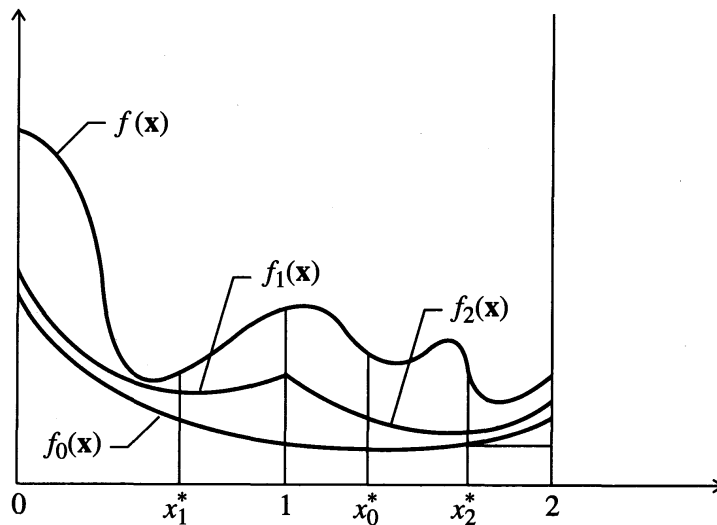
10.3 BRANCH-AND-BOUND METHODS

We have already discussed branch-and-bound methods in Sections 9.2 and 9.3 in the context of mixed-integer linear and nonlinear programming. The “divide-and-conquer” principles underlying BB can also be applied to global optimization without discrete variables. The approximation procedure for a function of one variable is shown in Figure 10.2. The nonconvex function f has three local minima over the interval $[0, 2]$. The convex underestimator function $f_0(x)$ is defined over the entire interval. The underestimating functions $f_1(x)$ and $f_2(x)$ are defined over the two subintervals, and are “tighter” underestimates than f_0 . We will discuss procedures for constructing such functions shortly. Because each underestimator is convex, minimizing it using any convergent local solver leads to its global minimum. Let x_i^* minimize $f_i(x)$ over its associated interval, as shown in Figure 10.2. Then $f_i(x_i^*)$ is a lower bound on the global minimum over that interval, and $f(x_i^*)$ is an upper bound over the entire interval. These bounds are used to fathom nodes in the BB tree, in the same way as LP relaxations were used in Chapter 9.

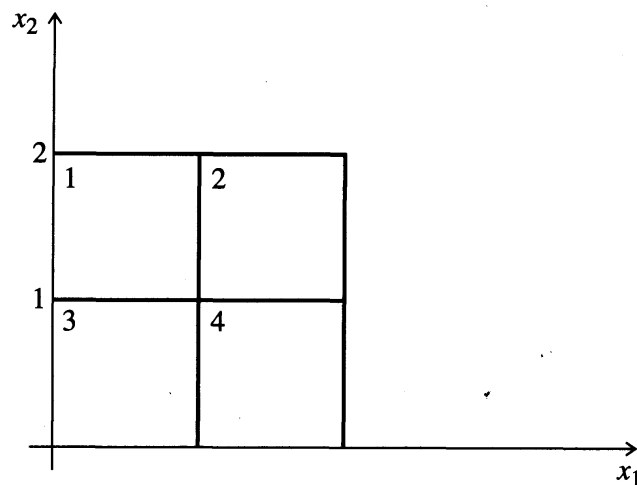
To illustrate, consider a minimization problem involving two variables with upper and lower bounds:

$$\text{Minimize: } f(\mathbf{x})$$

$$\text{Subject to: } 0 \leq x_i \leq 2, \quad i = 1, 2$$

**FIGURE 10.2**

Convex underestimator of a nonconvex function.

**FIGURE 10.3**

Branch-and-bound partitions.

where $\mathbf{x} = (x_1, x_2)$ and f is a nonconvex function having several local minima within the rectangle defined by the bounds. Let the initial partition be composed of four smaller rectangles, as shown in Figure 10.3.

Figure 10.4 shows a BB tree, with the root node corresponding to the original rectangle, and each node on the second level associated with one of these four partitions. Let $f_i(x)$ be the underestimating function for the partition associated with node i . The lower bounds shown next to each node are illustrative and are derived by minimizing $f_i(x)$ over its partition using any local solver, and the upper bounds

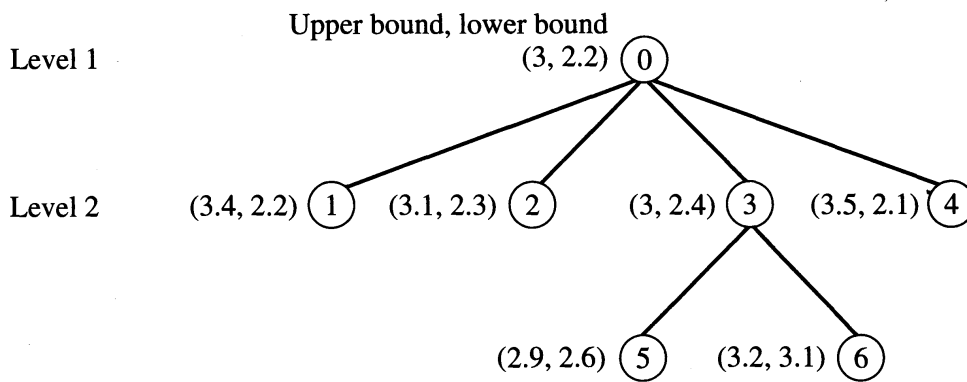


FIGURE 10.4
Branch-and-bound tree.

are the f values at the points that minimize f_i . Because the f_i functions for each partition are better estimates for f over their rectangles than f_0 is, the lower bounds over each partition are generally larger than the lower bound at the root node, as shown in Figure 10.4. The upper bounds need not improve at each level, but the f and \mathbf{x} values associated with the smallest upper bound found thus far are retained as the “incumbent,” the best point found thus far. The best f value at level two is 3.0.

The iterative step in this BB procedure consists of choosing a node to branch on and performing the branching step. There are several rules for choosing this node. A popular rule is to select the node with the smallest upper bound, on the assumption (possibly incorrect) that it leads to better f values sooner. This is node 3 in Figure 10.4. This node’s rectangle is partitioned into two (or possibly more) subsets, leading to nodes 5 and 6, and the convex subproblems at each node are solved, yielding the upper and lower bounds shown. Because the lower bound at node 6 (3.1) exceeds the incumbent value of 3.0, an f value lower than 3.0 cannot be found by further branching from node 6. Hence this node has been fathomed. The procedure stops when the difference between the incumbent f value and the lower bound at each unfathomed node is smaller than a user-defined tolerance.

If each underestimating function is “tighter” than that at the node immediately above in the tree, the BB procedure eventually terminates. Floudas (2000a) suggests procedures for constructing these underestimating functions, which apply to specific commonly occurring nonconvex functions, such as the bilinear function xy , quotients x/y , concave functions of a single variable, and so on. For a general function $f(\mathbf{x})$ of n variables, defined over a rectangle $\mathbf{l} \leq \mathbf{x} \leq \mathbf{u}$, a convex underestimator is

$$L(\mathbf{x}) = f(\mathbf{x}) + \sum_{i=1}^n \alpha_i (l_i - x_i)(u_i - x_i) \quad (10.4)$$

where the α_i ’s are positive scalars. Because the summation in Equation (10.4) is nonpositive, $L(\mathbf{x}) \leq f(\mathbf{x})$, so L is an underestimator of f . The summation is a positive linear combination of convex quadratic functions, so this term is convex. For L to

be convex, the scalars α_i must be sufficiently large, as can easily be seen by considering the Hessian matrix of L :

$$\nabla^2 L(\mathbf{x}) = \nabla^2 f(\mathbf{x}) + \mathbf{D} \quad (10.5)$$

where \mathbf{D} is a positive diagonal matrix with diagonal elements $2\alpha_i$. If these elements are sufficiently large, the Hessian of L is positive-definite for all x in the domain of f , which implies that L is convex over that domain. Tests to determine how large the α_i 's need to be are in Floudas (2000a, b), and other references cited there. Floudas has also shown that the maximum difference between the approximating function L and the actual function f is

$$d_{\max} = \frac{1}{4} \sum_{i=1}^n \alpha_i (u_i - l_i)^2 \quad (10.6)$$

As rectangles are partitioned, the difference $(u_i - l_i)$ decreases, so the successive underestimating functions become tighter approximations to f .

10.4 MULTISTART METHODS

Because software to find local solutions of NLP problems has become so efficient and widely available, *multistart methods*, which attempt to find a global optimum by starting the search from many starting points, have also become more effective. As discussed briefly in Section 8.10, using different starting points is a common and easy way to explore the possibility of local optima. This section considers multistart methods for unconstrained problems without discrete variables that use randomly chosen starting points, as described in Rinnooy Kan and Timmer (1987, 1989) and more recently in Locatelli and Schoen (1999). We consider only unconstrained problems, but constraints can be incorporated by including them in a penalty function (see Section 8.4).

Consider the unconstrained global minimization of a smooth function of n variables $f(\mathbf{x})$. We assume that upper and lower bounds can be defined for each variable, so that all local minima lie strictly inside the rectangle R formed by the bounds. Let L denote the local optimization procedure to be used. L is assumed to operate as follows: Given any starting point, \mathbf{x}_0 in R , L converges to a local minimum of f that depends on \mathbf{x}_0 and is "closest" to \mathbf{x}_0 in a loose sense. If L is started from each of N randomly generated starting points, which are uniformly distributed in R , the probability that the best local optimum found in these N trials is global approaches one as N approaches infinity (Rinnooy Kan and Timmer, 1989). In fact, L is not even necessary for this asymptotic result to hold, because the best function value over all the starting points converges to the globally optimal value as $N \rightarrow \infty$ with probability 1, but this search usually converges much more slowly.

Because each local optimum may be found many times, this multistart procedure is inefficient. If \mathbf{x}_i^* denotes the i th local optimum, we define the *region of*

attraction of \mathbf{x}_i^* , R_i , to be the set of all starting points in R from which L converge to \mathbf{x}_i^* . The goal of an efficient multistart method is to start L *exactly once* in each region of attraction.

Rinnooy Kan and Timmer (1987, 1989) developed an efficient multistart procedure called multilevel single linkage (MLSL), based on a simple rule. A uniformly distributed sample of N points in R is generated, and the objective function f is evaluated at each point. The points are sorted according to their f values, and the pN best points are retained, for which p is an algorithm parameter between 0 and 1. L is started from each point of this reduced sample, except if another sample point with a lower f value exists within a certain critical distance. L is also not started from sample points that are too near the boundary of R or too close to a previously discovered local minimum. Then, N additional uniformly distributed points are generated, and the procedure is applied to the union of these points and those retained from previous iterations. The critical distance decreases each time a new set of sample points is added. The authors show that, even if the sampling continues indefinitely, the total number of local searches ever initiated by MLSL is finite with a probability of 1, and each local minimum of f is located with a probability of 1. They also developed Bayesian stopping rules, which incorporate assumptions about the costs and potential benefits of further function evaluations, to determine when to stop the procedure.

10.5 HEURISTIC SEARCH METHODS

Chapter 9 describes several types of problems that require the use of integer-valued variables and discusses two solution approaches for such problems: branch-and-bound (BB) and outer approximation (OA). These methods can guarantee a global solution under certain conditions, but the computational effort required increases rapidly with the number of integer variables. In addition, BB is guaranteed to find a global optimum only if the global optimum of each relaxed subproblem is found. As discussed in Sections 10.3 and 10.4, this may be very hard to do if the subproblems are not convex. OA also requires convexity assumptions to guarantee a global solution (see Section 9.4). Hence, there is a need for alternative methods that are not guaranteed to find an optimal solution, but often find good solutions more rapidly than BB or OA. We describe such methods, called heuristic search procedures, in this section. They include genetic algorithms (or, more generally, evolutionary algorithms), simulated annealing, tabu search, and scatter search.

Heuristic search procedures can be applied to certain types of combinatorial problems when BB and OA are difficult to apply or converge too slowly. In these problems, it is difficult or impossible to model the problem in terms of a vector of decision variables, which must satisfy bounds on a set of constraint functions, as required by OA. One example is the “traveling salesman” problem, in which the feasible region is the set of all “tours” in a graph, that is, closed cycles or paths that visit every node only once. The problem is to find a tour of minimal distance or cost,

which is to be used by a vehicle that is routed to several stops. The traveling salesman problem is the simplest type of vehicle-routing problem, with a single vehicle leaving from a single starting point. Multivehicle, multistarting point problems can have constraints such as time windows within which stops must be visited, vehicle capacities, restrictions on which vehicles can visit which stops, and so on (Crainic and Laporte, 1998). In all cases, the problem is to find a set of routes and an assignment of vehicles to routes, that visit all stops and meet all the constraints.

Another important class of combinatorial problems is “job-shop” scheduling, in which you seek an optimal sequence or order in which to process a set of jobs on one or more machines. Such problems are often encountered in chemical engineering when sequencing a set of products through a batch process, in which set-up times and costs must be incurred for each unit operation before a product can be produced, and these times depend on the product previously produced. If there is a single machine, and the products are numbered from 1 to n , then the feasible region is the set of all permutations of the positive integers 1, 2, ..., n , corresponding to the order in which the jobs are processed. This is equivalent to a traveling salesman problem in which each node in a graph corresponds to a job, and the travel times between nodes are the set-up times between jobs.

These combinatorial problems, and many others as well, have a finite number of feasible solutions, a number that increases rapidly with problem size. In a job-shop scheduling problem, the size is measured by the number of jobs. In a traveling salesman problem, it is measured by the number of arcs or nodes in the graph. For a particular problem type and size, each distinct set of problem data defines an *instance* of the problem. In a traveling salesman problem, the data are the travel times between cities. In a job sequencing problem the data are the processing and set-up times, the due dates, and the penalty costs.

One measure of the efficiency of an algorithm designed to solve a class of combinatorial problems is an upper bound on the time required to solve any problem instance of a given size, and this time increases with size. Time is often measured by the number of arithmetic operations or constraint and objective function evaluations to find a solution. If, for a given algorithm and problem class, it can be shown that the time required for the algorithm to solve any instance of the problem is bounded by a polynomial in the problem size parameter(s), then that algorithm is said to solve the problem class in *polynomial time*. Some combinatorial problems, for example, sorting a list, are solvable in polynomial time. For many combinatorial problems, however, no known algorithm can solve all instances in polynomial time. Such problems are called *NP-hard*. Although methods to find optimal solutions have been devised that avoid complete enumeration of all solutions (often based on branch-and-bound concepts), none of them can guarantee a solution in polynomial time. Hence, heuristic and metaheuristic search methods, which cannot guarantee an optimal solution but often find a good (or optimal) one quickly, are now widely used.

10.5.1 Heuristic Search

Consider the problem: Minimize $f(\mathbf{x})$ subject to $\mathbf{x} \in X$, where \mathbf{x} represents the variables or other entities over which we are optimizing f . The objective function f may

TABLE 10.1
Data for sequencing problem

Job	Processing time (days)	Due date (day)	Tardiness penalty (\$/day)
1	2	5	1
2	6	7	2
3	4	9	4

TABLE 10.2A
Objective function computation for the sequence (3, 1, 2)

Job	Completion time (days)	Tardiness (days)	Delay cost (\$)
1	$4 + 2 = 6$	$6 - 5 = 1$	$1 * 1 = 1$
2	$6 + 6 = 12$	$12 - 7 = 5$	$5 * 2 = 10$
3	4	$\text{Max}(4 - 9, 0) = 0$	$0 * 4 = 0$

TABLE 10.2B
Swap neighborhood of (3, 1, 2)

i	j	New permutation	Move value
1	2	(1, 3, 2)	$10 - 11 = -1$
1	3	(2, 1, 3)	$15 - 11 = 4$
2	3	(3, 2, 1)	$13 - 11 = 2$

be linear or nonlinear, and X is defined by the constraints of the problem. \mathbf{x} may be a cycle in a graph, a permutation (representing a sequence in which to process jobs on a machine), or, in the simplest case, a vector of n decision variables. The constraints may be bounds on functions of \mathbf{x} , or they may include verbal logic-based statements or conditions like “ x is a tree in this graph that connects all nodes” or “if-then” statements.

As an example, consider a problem of sequencing three jobs on a single machine to minimize the sum of weighted “tardiness” for all jobs, where tardiness is defined as the difference between the completion time of a job and its due date if this difference is positive, and zero otherwise. Job processing times, due dates, and delay penalties for an instance of this problem are shown in Table 10.1.

To show how the objective function is computed, consider the sequence $\mathbf{x} = (3, 1, 2)$. The job completion times, tardiness values, and delay costs for this sequence are shown in Table 10.2A.

The objective value for this sequence is the sum of the costs in the “delay cost” column:

$$f(3, 1, 2) = 11$$

TABLE 10.3
Descent method using the search
neighborhood $N(\mathbf{x})$

-
1. Start with $\mathbf{x} \in X$
 2. Find $\mathbf{x}' \in N(\mathbf{x})$ such that $f(\mathbf{x}') < f(\mathbf{x})$.
 3. If no such \mathbf{x}' exists, stop and return \mathbf{x} .
 4. Otherwise replace \mathbf{x} by \mathbf{x}' and return to step 2.
-

In neighborhood-based heuristic searches, each $\mathbf{x} \in X$ has an associated neighborhood $N(\mathbf{x})$ that contains all the feasible solutions that the search will explore when the current point is \mathbf{x} . Each alternative solution $\mathbf{x}' \in N(\mathbf{x})$ is reached from \mathbf{x} by an operation called a move. Consider again the three-job problem based on Table 10.2A. Let the current sequence \mathbf{x} be (3, 1, 2), and suppose that we consider only neighboring permutations \mathbf{x}' that can be reached from \mathbf{x} by swapping a pair of jobs in \mathbf{x} . This “swap neighborhood” is shown in Table 10.2B, in which i and j are the indices of the jobs to be swapped. If there are n jobs, then a swap neighborhood contains $n(n - 1)/2$ permutations.

The “move value” column in Table 10.2B contains the change in objective value realized by making the move, $f(\mathbf{x}') - f(\mathbf{x})$. Because $\mathbf{x} = (3, 1, 2)$, we determined earlier that $f(\mathbf{x}) = 11$. If the objectives for the new permutations shown in Table 10.2B are evaluated, move values can be obtained. By moving to permutation (1, 3, 2), we improve the objective by one unit. Then the same procedure can be applied at this new point.

This straightforward descent method can be generalized for discrete-variable problems as shown in Table 10.3 (Glover and Laguna, 1997, Chapter 2). This algorithm is similar to the algorithms for linear programs and continuous-variable nonlinear programs discussed in Chapters 6–8, where step 2 was conducted by choosing a search direction and performing a line search along that direction. The variation of this algorithm that seeks the $\mathbf{x}' \in N(\mathbf{x})$ with lowest f value is called steepest descent (see Chapter 6). Although this simple descent method solves some combinatorial problems from any starting point, for many important problems (routing and sequencing included) it usually stops at a nonoptimal point, which is often far from optimal. Such a point is called a local solution relative to the neighborhood $N(\mathbf{x})$. As noted by Glover and Laguna (1997), descent methods by themselves have had very limited success in solving hard combinatorial optimization problems, but they provide an underlying heuristic for a *metaheuristic* procedure to guide the search. The resulting metaheuristic algorithms have been widely and successfully used.

In fact, most metaheuristics do not require a preexisting heuristic. They simply require a way to define a neighborhood of any current solution, which contains alternative solutions as possible moves. For example, tabu search, which is discussed in the following section, includes strategies for operating directly with such neighborhoods. Some neighborhood structures allow solutions to be built up one

element at a time in a constructive way. For example, a spanning tree in a network may be constructed one arc at a time, each time choosing a new arc that creates no loops, connects a new node, and has the greatest value or least cost.

10.5.2 Tabu Search

Tabu search (TS) is widely used by operations research analysts, but has received little attention from chemical engineers, even though it can be used to solve many important and difficult real-world problems. These include problems of the following types: planning and scheduling, telecommunications and multiprocessor computing systems, transportation networks and vehicle routing, operation and design of manufacturing systems, and financial analysis. An excellent survey of these applications and pertinent references is found in Glover and Laguna (1997).

As discussed in the previous section, descent heuristics fail to solve many problems because they get trapped in local minima (relative to the type of neighborhoods they use). That is, they stop at the first solution encountered where no neighboring solution is better. TS, and in fact any metaheuristic search method, overcomes this limitation by allowing nonimproving moves. The term *tabu* refers to TS's definition of certain moves as forbidden. These are usually specified as moves to solutions with particular attributes, as illustrated in the following example. The tabu moves are specified so as to keep previously performed moves from being reversed or to prevent already visited solutions from being revisited. These and other mechanisms force the search process to move beyond the nearest local minimum and to explore regions where improved solutions may lie.

We explain the ideas behind TS using a problem from Barnes and Vanston (1981) of sequencing five different product batches through a single-batch process. Each batch has a processing time and a delay penalty cost, as shown in Table 10.4. The penalty is charged for any delay in starting production beyond time zero; set-up costs must also be taken into account.

It is reasonable to schedule jobs with short processing times and high penalty costs first. This is motivation for a heuristic that computes the ratio of processing time to penalty cost (see column four of Table 10.4) and sequences the batches in order of increasing value of this ratio, which is the order given in the table. However,

TABLE 10.4
Batch processing times and delay penalties

Batch	Processing time (h)	Delay penalty (100\$/h)	Ratio
1	3	7	$3/700 = 4.28\text{E-}3$
2	4	8	$4/800 = 5.0\text{E-}3$
3	1	1	$1/100 = 1.0\text{E-}2$
4	4	3	$4/300 = 1.33\text{E-}2$
5	5	2	$5/200 = 2.5\text{E-}2$

TABLE 10.5
Batch set-up costs

		$j \rightarrow$					
i		1	2	3	4	5	6
↓	0	11	6	12	20	14	
	1		13	7	12	11	10
	2	9		11	13	6	12
	3	9	10		20	7	15
	4	10	7	8		6	12
	5	14	13	12	13		9

TABLE 10.6
Calculated completion times

Batch	Completion time
1	3
2	$4 + 4 = 8$
3	$3 + 1 = 4$
4	$13 + 4 = 17$
5	$8 + 5 = 13$

this ignores the fact that, if batch i was last produced, and batch j is next, there is a set-up cost of $s_{i,j}$ dollars before batch j can begin, representing the time and expense associated with cleaning up after batch i and preparing the process to produce batch j . These set-up costs are shown in Table 10.5. The table includes fictitious batches 0 and 6 (always sequenced first and last, respectively, and with zero processing times and delay penalties), whose set-up costs represent the cost of starting up the first batch and cleaning up after the last one.

Let

$$P = (p(1), p(2), \dots, p(5))$$

be a permutation of the integers 1 through 5, representing a sequence for producing the batches, where $p(i)$ is the index of the job in position i . If $P = (1, 3, 2, 5, 4)$, then the completion times of the jobs are as shown in Table 10.6.

The corresponding objective value $obj(P)$ is computed as follows:

$$\text{Objective}(P) = \text{Delay cost}(P) + \text{Setup cost}(P)$$

$$\text{Delay cost}(P) = 700(3) + 800(8) + 100(4) + 300(17) + 200(13) = 16,600$$

$$\text{Setup cost}(P) = 1100 + 700 + 1000 + 600 + 1300 + 1200 = 5900$$

$$\text{Objective}(P) = 22,500$$

A TS algorithm for this problem described in Laguna, et al. (1991) modifies the swap heuristic as follows:

- At each iteration, certain moves are forbidden or *tabu*.
- One or more move *attributes* are chosen, and the tabu moves are those whose attribute(s) satisfy the specified tabu conditions.
- A *short-term memory function* determines how long a tabu restriction remains active. This can be expressed as the number of iterations a tabu condition is enforced once it is imposed.
- The tabu status of a move can be overridden if the objective value after the move is better than a specified threshold, called an *aspiration level*.
- A *long-term memory function* determines when to restart the entire procedure and what the new starting point should be. These new starting points are chosen to be in regions of the search space (i.e., the space of all permutations) that have not been previously explored. This *diversifies* the search. Long-term memory can diversify the search in ways other than by direct restarting (Glover and Laguna, 1997) and can also *intensify* the search by inducing it to explore attractive areas more thoroughly.

The purpose of a tabu restriction is to prevent a move from being reversed during the length of the short-term memory, which is a number of future moves specified by the variable *tabu_size*. If, at a given iteration, jobs $p(i)$ and $p(j)$ are swapped, then any move that places job $p(i)$ *earlier* in the sequence than position i is tabu, until *tabu_size* iterations have occurred or the aspiration level is exceeded. To keep track of which moves are tabu and to free those moves from their tabu status, Laguna et al. define the following data structures

- $\text{tabu_list}(k) = p(i)$ if job $p(i)$ is prevented from moving to the left of its tabu position at iteration k . This is a circular list of length *tabu_size*.
- $\text{tabu_position}(p(i)) = \text{tabu position for job } p(i)$.
- $\text{tabu_state}(p(i)) = \text{number of times job } p(i) \text{ appears on the tabu list}$.
- $\text{aspiration_level}(p(i)) = \text{aspiration level for job } p(i)$.

The aspiration level allows a tabu move of a job $p(j)$ to an earlier position if

$$\text{Current objective value} + \text{move_value} < \text{aspiration level for job } p(j)$$

The aspiration level for a job is initialized to a large value and updated as follows. Let P be the current sequence and assume that the move of jobs $p(i)$ and $p(j)$ has the best *move_value*.

- If $\text{aspiration level}(p(j)) > \text{objective}(P)$, then $\text{aspiration level}(p(i)) = \text{objective}(P)$
- If $\text{aspiration level}(p(j)) > \text{objective}(P) + \text{move_value}$, then $\text{aspiration level}(p(j)) = \text{objective}(P) + \text{move_value}$.

This prevents the immediate reversal of a nonimproving move (one with a positive *move_value*) in the next iteration. The reversal of this move now has a negative *move_value*, but it is classified as tabu, and the previous update does not allow it to satisfy the aspiration criterion.

Begin

- Initialize long term memory function
- Best_obj = large value
- **Do while** (Best_obj has changed in the last max_moves_long starting points) **Begin1**
 - Generate starting solution P , and set Best_solution = P
 - Evaluate $obj(P)$
 - Initialize move_value_matrix
 - Initialize short term memory function
 - Do while** (moves without improvement < max_moves) **Begin2**
- Update long term memory function
- Best_move value = large value
- **For** (all candidate moves) **Begin3**
- **If** (candidate move is admissible) **Begin4**
- **If** (move_value < best_move value) **Begin5**
- Best_move value = move_value
- Best_move = current_move **End5**
- End4**
- End3**
- Execute best_move
- Update objective value: $obj(P) = obj(P) + best_move_value$
- Update move_value_matrix
- Update short term memory function
- **If** $obj(P) < Best_obj$, then **Begin6**
- Best_obj = $obj(P)$
- Best_solution = P **End6**
- End2**
- End1**

FIGURE 10.5

Tabu search procedure for batch sequencing.

Figure 10.5 shows the TS procedure in pseudo-code. This entire procedure is executed until max_moves_long successive restarts fail to improve the best objective value. Given a starting solution, the inner **do** loop is executed until there are max_moves successive moves without improvement in the best solution found in the current “pass,” that is, using the current starting solution. A move is considered a *candidate* if the jobs being swapped are within a specified “distance” (number of positions) of one another. This limitation allows search time to be limited, but a complete search can be done by making this distance equal to the number of batches. A candidate move is admissible if either it is not tabu or it is tabu but its tabu status is overridden by the aspiration criterion.

The long-term memory function uses the matrix called “move_value_matrix” in Figure 10.5, whose (i, j) element is the number of times that job i has been scheduled in position j . This matrix is updated after every move by adding 1 to the (i, j) element if $p(i) = j$ in the current sequence. Then, the fraction of time each job has spent in each position can be calculated by dividing these matrix elements by the

total number of moves so far. Penalty costs proportional to these time fractions are defined and are used in the heuristic that generates starting solutions to force it to choose diverse starting points. This one-pass heuristic starts by scheduling batch 0 in position 0. Then, the unsequenced job j that minimizes the “distance” from the previously selected job, say job i , is scheduled next. The “distance” is the set-up cost between jobs i and j plus a multiple of the ratio of the delay penalty for job j divided by the largest delay penalty for all unsequenced jobs. If this heuristic is being used to restart the algorithm, a multiple of the fraction of time that job j has occupied the current position is added to the “distance,” biasing it to choose different positions for the jobs from those they occupied frequently thus far. Such diversification strategies are important elements of intelligent search procedures.

The performance of this TS algorithm on the five-batch problem described earlier is shown in Table 10.7, using the following TS parameter values:

- Maximum moves without improvement = `max_moves` = 2.
- Maximum number of positions between swapped jobs = 1.
- Length of short-term memory = `tabu_size` = 3.
- Maximum restarts without improvement = `max_moves_long` = 4.

At iteration 1, the best move interchanges jobs 1 and 3, with a move value of -1000 . This leads to the new sequence in row 2. In row 1, because job 3 was moved to the right, it is added to `tabu_list` in its first position, `tabu_state` (3) is set to 1 because job 3 appears once on `tabu_list`, and `tabu_position` (3) is set to 1, the original position of job 3. Moves that swap job 3 back to position 1 are henceforth tabu, unless they satisfy the aspiration criterion. At iteration 2, the best available move is to swap jobs 2 and 3, so job 3 is again added to `tabu_list`, its `tabu_state` is increased to 2, and its `tabu_position` entry is changed to 2. Iterations 2 and 3 fail to improve `Best_obj`, so the inner loop is restarted at iteration 4. Note that the current schedule in row 4 is quite different from those in earlier rows, due to the long-term memory function. The schedule in row 5 is optimal, but there is no way to prove its optimality, so the search must continue. It is restarted at iterations 7, 9, and 11, and the procedure stops at iteration 12 due to the limit of four successive restarts without improvement. A linear mixed-integer programming formulation of a similar production sequencing problem is described in Chapter 16.

Unfortunately, no general-purpose TS software is commercially available. Thousands of TS implementations have been made over the last 15 years (Glover and Laguna, 1997), but all address specific classes of problems, such as the job sequencing problem discussed earlier. Many of these implementations have been extremely successful, because the flexibility of TS allows an experienced analyst to incorporate his or her knowledge of the problem into the algorithm in many ways. Specific knowledge can include selecting the neighborhood that defines the possible next solutions, the short- and long-term memory structures, and the attributes that determine which solutions are tabu, among other things.

In closing this section, we emphasize that the adaptive memory structures used in TS encompass a variety of elements not treated in this simple example. Further details can be found in Glover and Laguna (1997).

TABLE 10.7
Performance of tabu search on a five-batch sequencing problem

Iteration	Current schedule	Current objective	Best move	Move value	Tabu state	Tabu list	Tabu position	Best objective
0*	(3,1,2,4,5)	14900			(0,0,0,0,0,3)	(6,6,6)	(0,0,0,0,0)	14900
1	(3,1,2,4,5)	14900	(3,1)	-1000	(0,0,1,0,0,2)	(3,6,6)	(0,0,1,0,0)	
2	(1,3,2,4,5)	13900	(3,2)	1000	(0,0,2,0,0,1)	(3,3,6)	(0,0,2,0,0)	13900
3	(1,2,3,4,5)	14900	(1,2)	-900	(1,0,2,0,0,0)	(3,3,1)	(1,0,2,0,0)	
4*	(2,4,1,3,5)	15500	(4,1)	-2000	(0,0,0,1,0,2)	(4,6,6)	(0,0,0,2,0)	
5	(2,1,4,3,5)	13500	(4,3)	500	(0,0,0,2,0,1)	(4,4,6)	(0,0,0,3,0)	13500
6	(2,1,3,4,5)	14000	(1,3)	0	(1,0,0,2,0,0)	(4,4,1)	(2,0,0,3,0)	
7*	(3,2,1,5,4)	16500	(5,4)	-1600	(0,0,0,0,1,2)	(5,6,6)	(0,0,0,0,4)	
8	(3,2,1,4,5)	14900	(3,2)	-900	(0,0,1,0,1,1)	(5,3,6)	(0,0,1,0,4)	
9*	(3,1,2,5,4)	15900	(3,1)	-1000	(0,0,1,0,0,2)	(3,6,6)	(0,0,1,0,0)	
10	(1,3,2,5,4)	14900	(5,4)	-1000	(0,0,1,0,1,1)	(3,5,6)	(0,0,1,0,4)	
11*	(3,1,4,2,5)	16200	(4,2)	-1300	(0,0,0,1,0,2)	(4,6,6)	(0,0,0,3,0)	
12	(3,1,2,4,5)	14900	(3,1)	-1000	(0,0,1,1,0,1)	(4,3,6)	(0,0,1,3,0)	

*Current solution is a new starting point.

10.5.3 Simulated Annealing

Simulated annealing (SA) is a class of metaheuristics based on an analogy to the annealing of metals. Consider a solid with crystalline structure being heated to a molten state and then cooled until it solidifies. If the temperature is reduced rapidly, irregularities appear in the crystal structure of the cooling solid, and the energy level of the solid is much higher than in a perfectly structured crystal. If the material is cooled slowly, with the temperature held steady at a series of levels long enough for the material to reach thermal equilibrium with its environment, the final energy level will be minimal. Let the state of the system at any temperature be described by a vector of coordinates \mathbf{q} . At a given temperature, while the system is attaining equilibrium, the state changes in a random way, but transitions to states with lower energy levels are more likely at lower temperatures than at higher ones.

To apply these ideas to a general optimization problem, let the system state vector \mathbf{q} correspond to the objects to be optimized (job sequences, vehicle routes, or vectors of decision variables), denoted by \mathbf{x} . The system energy level corresponds to the objective function $f(\mathbf{x})$. As in Section 10.5.1, let $N(\mathbf{x})$ denote a neighborhood of \mathbf{x} . The following procedure (Floquet et al., 1994) specifies a basic SA algorithm:

- Choose an initial solution \mathbf{x} , an initial temperature T , a lower limit on temperature $TLOW$, and an inner iteration limit L .
- While ($T > TLOW$), do
 - For $k = 1, 2, \dots, L$, do
 - Make a random choice of an element $\mathbf{x}' \in N(\mathbf{x})$.
 - $Move_value = f(\mathbf{x}') - f(\mathbf{x})$
 - If $move_value \leq 0$ (downhill move), set $\mathbf{x} = \mathbf{x}'$
 - If $move_value > 0$ (uphill move), set $\mathbf{x} = \mathbf{x}'$ with probability $\exp(-move_value/T)$.
 - End inner loop
- Reduce temperature according to an *annealing schedule*. An example is $new\ T = cT$, where $0 < c < 1$.
- End temperature loop

Simulated annealing depends on randomization to diversify the search, both in selecting a move to evaluate (all moves to neighboring points are equally likely) and in deciding whether or not to accept a move. This basic SA algorithm uses the *Metropolis algorithm* (Johnston et al., 1989) to determine move acceptance, in which downhill moves are always accepted and uphill moves are accepted with a probability $\exp(-move_value/T)$. Note that, as T approaches 0, the probability of accepting an uphill move approaches 0. Hence, when the temperature is high, many uphill moves may be accepted, thereby possibly preventing the method from being trapped at a local minimum with respect to the neighborhood $N(\mathbf{x})$. The *Glauber algorithm* accepts all moves with the following probability:

$$\text{Glauber probability} = \frac{\exp(-move_value/T)}{1 + \exp(-move_value/T)}$$

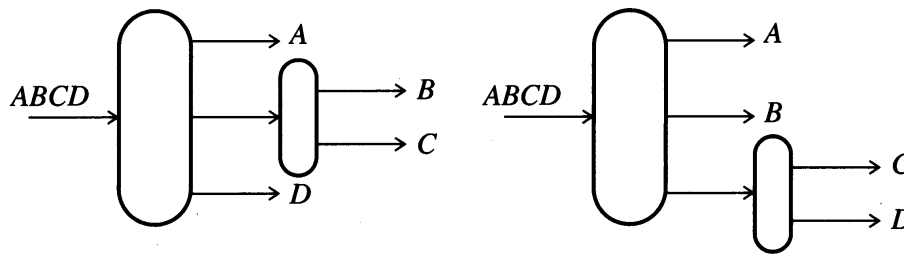


FIGURE 10.6
Separation sequences.

so here an improving move may be rejected. This leads to a search that is well diversified, so it will come closer to a global optimum, but may take longer than a Metropolis-based search, which is more likely to find a good solution quickly.

Applying simulated annealing to separation sequence synthesis

Floquet et al. (1994) applied SA to problems of separating a mixture of n components into pure products at minimal annual investment plus operating costs. The assumptions used were

- Each component of the feed stream exits in exactly one output stream of a separator. This is called *sharp separation*.
- Only one input/two output (simple) or one input/three output (complex) sharp separators are used.

Under these assumptions, the problem is to select the separators to be connected and the way they will be connected. Two possible separation sequences are shown in Figure 10.6. Floquet et al. (1994) show how to encode possible separation sequences as vectors containing the entries $\{-1, 0, 1\}$, which satisfy appropriate restrictions, and how to transform such vectors into neighboring sequences. For example, some transformations correspond to the insertion or deletion of a complex separator. Given this definition of a solution \mathbf{x} and its neighborhood $N(\mathbf{x})$, and given fixed and operating costs for each type of separator that defines the objective function $f(\mathbf{x})$, the authors applied simulated annealing to find the cheapest separation sequence. In solving problems with 5, 10, and 16 components with known optimal solutions, their SA algorithm found optimal solutions for all cases, and less than 2% of the feasible sequences were evaluated when the best solution was found. Recall, however, that an optimal solution is not guaranteed in general, and there is no way to tell when an optimal solution has been found unless the optimal objective value is known in advance.

10.5.4 Genetic and Evolutionary Algorithms

With the exception of parallel implementations (which are becoming increasingly important), tabu search and simulated annealing operate by transforming a single solution at a given step. By contrast, genetic algorithms (GAs) work with a set of

solutions $P = \{x_1, x_2, \dots, x_p\}$, called a *population*, with each population member x_i , called an *individual* or *member*. An initial population is created, and the population at the start of an iteration is modified by replacing one or more individuals with new solutions, which are created either by combining two individuals (*crossover*) or by changing an individual (*mutation*). The procedure is inspired by the evolution of populations of living organisms, whose chromosomes undergo crossover and mutation during reproduction. The genetic algorithm template that follows corresponds to the description in Reeves (1997).

- Choose an initial population, and evaluate the fitness of each individual.
- While termination condition not satisfied **do**
 - **If** crossover condition satisfied **then**
 - Select parent individuals.
 - Choose crossover parameters.
 - Perform crossover.
 - **If** mutation condition satisfied **then**
 - Choose mutation points.
 - Perform mutation.
 - Evaluate fitness of offspring.
 - Update population.

We now discuss the main steps of this algorithm. For more details, see Reeves (1997) and several other articles in that issue of the *INFORMS Journal on Computing*.

Solution encoding

In the original genetic algorithms proposed by Holland (1975), the individuals were binary vectors that represented encodings of solutions. For example, if a solution \mathbf{x} is a vector of n decision variables, a binary encoding is obtained by representing each component of \mathbf{x} as a binary number and concatenating these bit strings. In this encoding, the bits 0 and 1 are called the *alphabet*. Other alphabets are possible, and many GAs are designed to deal with \mathbf{x} vectors of n variables directly without any encoding.

Initial population and population size

The initial population should be diverse. Elements are often generated randomly using a uniform distribution over the solution space. As for population size, many authors have reported satisfactory results with population sizes as small as 30, although values of 50–100 are more common.

Crossover and mutation conditions

Crossover and mutation conditions are usually randomized rules, which determine if these operators are to be applied in the current iteration. Crossover is commonly applied in most if not all iterations, whereas mutation is applied less frequently.

Crossover and mutation

The crossover operation replaces some of the elements in each parent solution with those in the other. For example, in *one-point crossover*, with parents $P1$ and $P2$ represented by real-valued vectors, and with the crossover point after the third component, the parents and offspring are as shown here for a five-variable problem:

$$P1 = (1.2, 3, 5, 3.1, 4) \quad O1 = (1.2, 3, 5, 6.3, 5)$$

$$P2 = (2, 1, 0, 6.3, 5) \quad O2 = (2, 1, 0, 3.1, 4)$$

Multipoint crossover is also used, with r crossover points chosen randomly. Crossover can be further generalized by making r a random variable, and copying an element from the first parent with probability q , and from the second parent with probability $(1 - q)$. The case $q = 0.5$ is called *uniform crossover*. As an example of a mutation operator, for populations of real-valued vectors, Fogel (1995) suggests simply adding a Gaussian random variable to each component of a population member. When the individuals are bit strings, the “mutation points” are often randomly selected bits, which are then complemented to create the new solution.

In an *evolutionary algorithm*, the “classical” crossover operation is replaced by a more general “recombination” operation, which can be any procedure that combines two or more “parents” to produce one or more “offspring.” As an example, the scatter search procedure described in Glover and Laguna (1997) uses linear combinations of several individuals to produce offspring. Fogel (1995) creates one offspring from each individual (a vector of n real numbers) by adding an independent, normally distributed random variable to each component. This can also be viewed as a replacement for mutation.

Fitness and its role in selecting parents and mutation candidates

In unconstrained optimization problems, you can use the value of the objective $f(\mathbf{x})$ as a measure of the “fitness” of an individual \mathbf{x} , but some transformation must be applied when the objective is being minimized (for example, use $-f(\mathbf{x})$). More generally, fitness can be any monotonically increasing function of the objective. Using the objective directly or some simple modification of it is rarely effective, however, because it is sensitive to objective function scaling. Consider two values of f : 10 and 20. Adding 1000 to f transforms these values to 1010 and 1020, whose percentage difference is much smaller. If the probability of being chosen to be a parent is equal to an individual’s share of total population fitness, then before adding 1000, these probabilities are $1/3$ and $2/3$, and after they are $1010/2030$ and $1020/2030$, both close to 0.5. Reeves (1997) recommends ranking procedures, the simplest of which ranks individuals in order of their objective function values and sets fitness equal to that ranking. Once a measure of fitness has been chosen, a common procedure for selecting parents or mutation candidates is random selection from the population, using a probability distribution that assigns higher probabilities to individuals with higher fitness, such as that used in the previous example.

Updating the population

After a number of new solutions are produced by crossover (or more generally, recombination) and mutation operations, improved solutions must be incorporated into the population. The best solution found thus far is almost always retained. A common strategy replaces a certain fraction of the remaining individuals, either with improved offspring or with new individuals chosen to maintain diversity. Another strategy is *tournament selection*, in which new solutions and current population members compete in a “tournament.” Each solution competes with K other solutions, which may be randomly selected, and, in each pairwise comparison, the solution with best fitness value wins. If P is the population size, the P solutions with the most wins become the new population.

Constraints

When there are constraints, GAs face a fundamental difficulty, namely that many crossover or mutation operators rarely yield feasible offspring, even if the parents are feasible. This can lead to a population with an excessive number of infeasible solutions. To alleviate this problem, GAs often include a penalty function in f (see Section 8.4) to measure fitness. A value must be chosen for the penalty weight, however. If this is too small, the original problem of too many infeasible solutions remains, and if it is too large, the search tends to reject points with small infeasibilities, even if they are close to an optimal solution.

For an excellent introduction to genetic algorithms, see the website constructed by Marek Obitko of the Czech Technical University at <http://cs.felk.cvut.cz/~xobitko/ga/>. It contains a genetic algorithm, coded as a Java Applet, which the user can run interactively, specifying his or her own objective if desired.

10.5.5 Using the Evolutionary Algorithm in the Premium Excel Solver

An evolutionary algorithm is included in the current release of Frontline Systems' Premium Excel Solver (for current information, see www.frontsys.com). It is invoked by choosing “Standard Evolutionary” from the Solver dropdown list in the Solver Parameters Dialog Box. The other nonlinear solver is “Standard GRG Non-linear,” which is the GRG2 solver described in Section 8.7. As discussed there, GRG2 is a gradient-based local solver, which will find the “nearest” local solution to its starting point. The evolutionary solver is much less likely to stop at a local minimum, as we illustrate shortly.

The “Options” box for the evolutionary solver is shown in Figure 10.7. The solver stops when either the time or iterations limit is reached, or when 99% of the population members have fitness values such that the fractional deviation between largest and smallest is less than the “Convergence” tolerance shown in the figure. The population size cannot be less than 10 or more than 200, and the initial population is chosen mainly by random sampling from within the hyperrectangle specified by the bounds on the variables. You are advised to define bounds for all variables, so the initial sampling can be performed from a hyperrectangle of limited

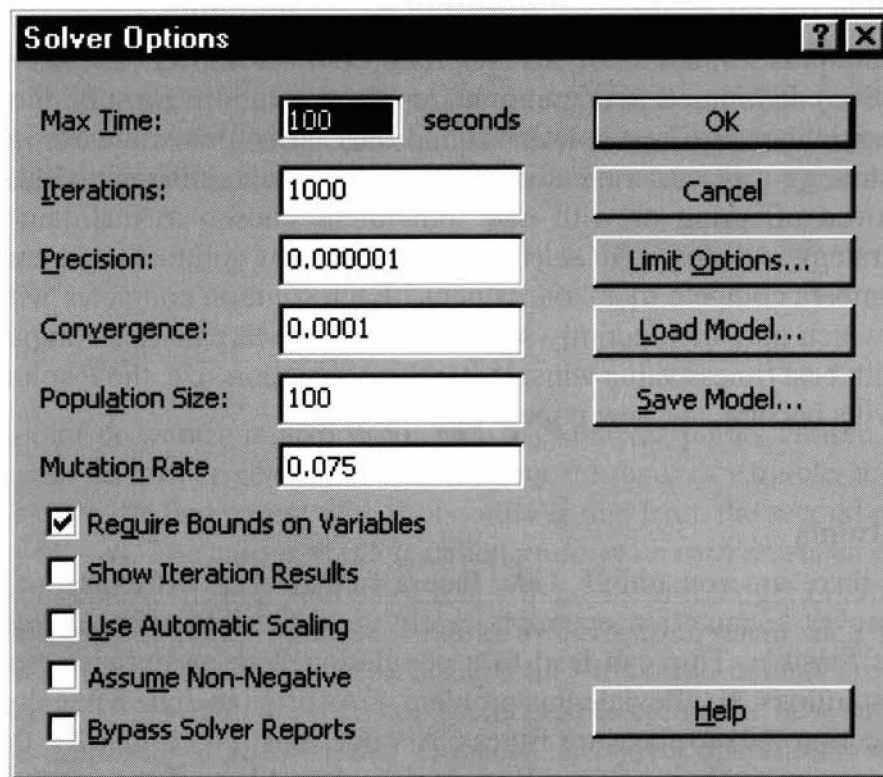


FIGURE 10.7

Options dialog for the evolutionary solver. Permission by Microsoft.

size. The initial decision variable values entered in the spreadsheet are also included in the initial population, perhaps several times, so the method benefits from a good starting point.

As in the GA template presented earlier, an iteration of the evolutionary algorithm consists of a crossover step involving two or more parents, mutation of a single population member (which is performed with the probability specified in the “Mutation Rate” box), and an optional local search. Note that the default mutation probability is 0.075, so if this value is used, mutation is fairly rare. Three mutation strategies are possible, one of which is selected if mutation is performed. A single variable in the single population element is selected for mutation. The three strategies alter the variables value as follows: (1) replace it by a random value from a uniform distribution; (2) move it to either its upper or lower bound; or (3) increase or decrease it by a randomly chosen amount, whose magnitude decreases as the iterations progress. In the population update, if a new element is “worse” than all population members, it is discarded. If not, the member to be replaced may not be the worst. Instead, a probabilistic replacement process is used, where the worst members have higher probabilities of being replaced. Computational experiments have shown that this leads to a more diverse population and to overall better performance than if the worst element were replaced each time. The measures of goodness used to define better and worse are complex, involving both objective values and penal-

TABLE 10.8
GRG results for Branin problem

Changing cells	Starting point 1	Starting point 2	Starting point 3
Initial x_1	1	-5	-5
x_2	1	5	10
Final x_1	3.141590675	9.000272447	-2.619502503
x_2	2.274999493	0.999727553	10
Final objective	0.397887358	2.550824843	2.791184064

ties for infeasibility. In some cases, infeasible points with good objective function values are accepted into the population. In others, an attempt is made to modify a solution to “repair” infeasibilities.

Table 10.8 shows the result of applying the “Standard GRG Solver” to a two-variable, one-constraint problem called the Branin problem that has three local optima and a global optimum with objective function value of 0.397. The objective function is constructed in three steps:

$$\begin{aligned}
 t_1 &= \left(\frac{x_1}{\pi} \right) \left[\left(\frac{1.275x_1}{\pi} \right) - 5 \right] \\
 t_2 &= (x_2 - t_1 - 6)^2 \\
 t_3 &= 9.602113 \cos(x_1) + 10 \\
 f &= t_2 + t_3
 \end{aligned} \tag{10.7}$$

and the problem is

$$\begin{aligned}
 &\text{Minimize: } f \\
 &\text{Subject to: } x_1 + x_2 \leq 10
 \end{aligned}$$

Starting from (1, 1), GRG finds the global solution, but it finds the two inferior local optima starting from the points (-5, 5) or (-5, 10). . . . The evolutionary solver finds the global optimum to six significant figures from any starting point in 1000 iterations.

This problem is very small, however, with only two decision variables. As the number of decision variables increases, the number of iterations required by evolutionary solvers to achieve high accuracy increases rapidly. To illustrate this, consider the linear project selection problem shown in Table 10.9. The optimal solution is also shown there, found by the LP solver. This problem involves determining the optimal level of investment for each of eight projects, labeled *A* through *H*, for which fractional levels are allowed. Each project has an associated net present value (NPV) of its projected net profits over the next 5 years and a different cost in each of the 5 years, both of which scale proportionately to the fractional level of investment. Total costs in each year are limited by forecasted budgets (funds available in

TABLE 10.9
Project selection problem with budget constraints

	Project								Optimal solution (Σ projects)	Funds available	
	A	B	C	D	E	F	G	H			
Net Present Value	151	197	119	70	130	253	165	300	Total NPV	839.11	
Costs:											
Year 1	20	100	20	30	50	40	50	80	Year 1	230	230
Year 2	20	10	10	30	10	20	40	30	Year 2	90	100
Year 3	20	0	10	30	10	20	10	20	Year 3	50	50
Year 4	20	0	10	20	10	20	10	0	Year 4	30	50
Year 5	10	30	10	10	10	20	10	0	Year 5	50	50
Optimal Decisions	0	0.667	0.222	0	0	1	0.778	1			

Abbreviation: NPV = net present value.

TABLE 10.10
Final objective values obtained by
evolutionary solver

Run	Iterations	Best objective value found
1	1,000	769.91
2	1,000	771.77
3	1,000	789.23
4	5,000	804.67
5	5,000	784.99
6	10,000	811.21
7	10,000	792.47
8	28,244	806.69

the last column), and the problem is to maximize the total value from all projects subject to the budget constraints. The summed NPV and the summed annual cost for each year at the optimum is in the next to last column. Note that the optimal solution in the bottom row of the table is at an extreme point because there are eight variables and eight active constraints (including those that require each decision variable to be between zero and one).

Table 10.10 shows the performance of the evolutionary solver on this problem in eight runs, starting from an initial point of zero. The first seven runs used the iteration limits shown, but the eighth stopped when the default time limit of 100 seconds was reached. For the same number of iterations, different final objective function values are obtained in each run because of the random mechanisms used in the mutation and crossover operations and the randomly chosen initial population. The best value of 811.21 is not obtained in the run that uses the most iterations or computing time, but in the run that was stopped after 10,000 iterations. This final value differs from the true optimal value of 839.11 by 3.32%, a significant difference, and the final values of the decision variables are quite different from the optimal values shown in Table 10.9.

If constraints that the decision variables be binary are added, however, the evolutionary solver reaches the optimal objective value of 767 in two runs with a 5000-iteration limit, and in one of two runs with a 1000-iteration limit. This is because only $2^8 = 256$ possible solutions need to be explored. Hence, if high accuracy is required, general-purpose evolutionary algorithms seem best suited to small problems with continuous variables, but they can find good solutions to larger problems, including integer and mixed-integer problems. Of course, evolutionary solvers (or any other search method) can be combined with local solvers like GRG, simply by starting the local solver at the final point obtained by the search procedure. If the problem is smooth and this point is near a global optimum, the local solver may well find the global solution to high accuracy. A local solver or heuristic can also be combined with scatter search, as described in the next section.

10.5.6 Scatter Search

Scatter search, described in Glover and Laguna (1997) and Glover (1998), is a population-based search method that primarily uses deterministic principles to strategically guide the search. Its steps are shown here, stated for problems whose only constraints are bounds on the variables. These bounds are taken into account when generating trial and combined solutions. It may, however, be applied to problems with more general constraints by augmenting the objective function with a penalty function (see Section 8.4).

Steps of Scatter Search

1. Create an initial diverse trial set of solutions.
2. Apply an *improvement method* to some or all trial solutions. Save the r best solutions found as members of the initial *reference set*, R .
3. Repeat steps 1 and 2 until some designated number of reference set solutions have been found.
4. Select subsets of the reference set to use in step 5.
5. For each subset chosen in step 4, use a *solution combination method* to produce one or more combined solutions.
6. Starting from each of the combined solutions in step 5, use the improvement method to create a set of enhanced solutions.
7. If an enhanced solution is better than any member of the reference set, insert it in the reference set and delete the worst member of the set.
8. Return to step 4, and repeat until some stopping condition is met. Such conditions may be based on elapsed time or iterations, or on lack of improvement in the objective.

Explanation of scatter search steps

Step 1 starts with a large set of “seed” solutions, which may be created by heuristics or by random generation. One possible implementation then generates a diverse subset of these by choosing some initial seed solution, then selecting a second one that maximizes the distance from the initial one. The third one maximizes the distance from the nearest of the first two, and so on.

The improvement method used in steps 2 and 6 may be one of the following:

- A heuristic descent method like that outlined in Figure 10.5 if the problem is combinatorial.
- A local NLP solver like the GRG or SQP algorithms described in Chapter 8; in this case, the problem must be a constrained, possibly nonconvex problem with continuous variables.
- Simply an evaluation of the objective and constraint functions.

Steps 4 through 6 are the scatter search counterparts to the crossover and mutation operators in genetic algorithms, and the reference set corresponds to the GA

population. The solution combination method produces combined solutions that are linear combinations of those in the subsets produced in step 4. However, variables that are required to take on integer values are subjected to generalized rounding processes, that is, processes for which the rounding of each successive variable depends on the outcomes of previous roundings. In the simplest case, two subsets, each containing a single solution, are chosen, with one solution selected to have a good objective value (to intensify the search in the neighborhood of good solutions) and the second chosen to be far from the first (to diversify the search). In this case, taking linear combinations of these two solutions produces new ones that are on the line segment between *and beyond* the two “parent” solutions. These are then used as starting points for the improvement method.

Scatter search has been implemented in software called OPTQUEST (see www.opttek.com). OPTQUEST is available as a callable library written in C, which can be invoked from any C program, or as a dynamic linked library (DLL), which can be called from a variety of languages including C, Visual Basic, and Java. The callable library consists of a set of functions that (1) input the problem size and data, (2) set options and tolerances, (3) perform steps 1 through 3 to create an initial reference set, (4) retrieve a trial solution from OPTQUEST to be input to the improvement method, and (5) input the solution resulting from the improvement method back into OPTQUEST, which uses it as the input to step 7 of the scatter search protocol. The improvement method is provided by the user. We use the term *improvement* loosely here because the user can simply provide an evaluation of the objective and constraint functions.

Optimizing simulations

OPTQUEST has also been combined with several Monte Carlo and discrete-event simulators. The Monte Carlo simulators include an Excel add-on called Crystal Ball (see www.decisioneering.com). It allows a user to define a subset of spreadsheet cells as random input variables with specified probability distributions and to designate several output cells that depend on these inputs and on other nonrandom input cells. The program then samples a specified number of times from the input distributions, evaluates the output cells, and computes statistics and histograms of the distributions of each output cell. In an optimization, a set of (nonrandom) input cells are designated as decision cells, some statistic associated with an output cell (typically its mean) is selected to be the objective function, and other statistics of other outputs may be taken as constraints. OPTQUEST is then applied to vary the decision variables in order to optimize the objective subject to the constraints. For each trial solution suggested by OPTQUEST, a complete simulation is run, and the designated cell statistics are returned to OPTQUEST. As an example, one can minimize the average of total holding plus set-up cost in an inventory problem with random demand, by choosing an optimal reorder level and order quantity. In such problems, the average value returned by Crystal Ball is only an estimate of the true average, so it contains some random error, which can be reduced by using a larger sample size. OPTQUEST is able to process these noisy objective values and still return a good approximation to an optimal solution.

TABLE 10.11
OPTQUEST applied to problem in Table 10.8

Iteration	Best objective	x_1	x_2
1	27.7029	1	1
4	8.8072	8.00106	1.99894
19	0.471901	3.19623	1.98847
89	0.406846	3.18067	2.28509
150	0.401044	3.15053	2.3207
205	0.39855	3.14473	2.29735
335	0.398046	3.14732	2.26958
459	0.397898	3.14194	2.2716
565	0.397887	3.14159	2.2751

TABLE 10.12
OPTQUEST applied to problem in Table 10.9

Iteration	Best objective	A	B	E
1	0.00	0	0	0
3	300.00	0	0	0
10	565.00	1	0	1
35	604.76	0.13	0.67	0.89
67	609.51	0.89	0.53	0.72
78	721.47	0.07	0.40	0.78
80	730.94	0.08	0.48	0.77
82	742.77	0.09	0.58	0.76
84	757.56	0.10	0.70	0.75
159	766.88	0.09	0.72	0.77
161	769.21	0.09	0.72	0.78
1005	794.13	0.28	0.69	0.72
2084	794.35	0.28	0.69	0.72
2963	794.73	0.27	0.69	0.72
4024	794.82	0.27	0.69	0.72
4996	797.25	0.24	0.70	0.72
Optimal	839.11	0.00	0.67	0.00

OPTQUEST examples

Crystal Ball can deal with spreadsheets that contain no random variables, and OPTQUEST can be applied to deterministic optimization problems arising from such spreadsheets. Table 10.11 shows the performance of OPTQUEST applied to the two-variable, one-constraint problem defined in Equations (10.7), which was solved by an evolutionary algorithm in Section 10.5 to six-digit accuracy in 1000 iterations. OPTQUEST finds the same solution with similar effort.

Table 10.12 shows OPTQUEST's progress on the project selection LP, whose optimal solution is given in Table 10.9. Initial progress is rapid, but it slows rapidly after about 1000 iterations, and after 5000 iterations the best objective value found is 797.25, about 5% short of the optimal value of 839.11. The values of variables A ,

TABLE 10.13
Classification of metaheuristic
search procedures

Metaheuristic	Classification
Genetic algorithms	M/S/P
Scatter search	A/N/P
Simulated annealing	M/S/1
Tabu search	A/N/1

B , and E are also shown. Although B is reasonably near its optimal value, A and E are far from theirs. This performance is comparable to that of the evolutionary algorithm in the Extended Excel Solver, shown in Table 10.10. If the decision variables in this problem must be binary, however, then OPTQUEST finds the optimal solution, whose objective value is 767, in only 116 iterations. The evolutionary algorithm found this same optimal solution in one of two runs using 1000 iterations.

Classifying metaheuristics

Glover and Laguna (1997) classify metaheuristics according to a three-attribute scheme as shown in Table 10.13. In the first position, “A” denotes the use of adaptive memory, and “M” means memoryless. An important feature of tabu and scatter search is remembering attributes of past solutions to guide the search in an adaptive way, that is, the length and operation of the memory may vary as the search progresses. Genetic algorithms and simulated annealing are viewed as not having *adaptive* memory, although GAs do retain information on the past through the population itself. An “N” in the second position indicates that a systematic neighborhood search is used to find an improved solution, and “S” indicates that a randomized sampling procedure is employed. Although traditional GA and SA methods use random sampling, some recent SA and evolutionary algorithms either replace this with a neighborhood search or initiate a search from a point found by a randomized procedure. A “1” in the third position indicates that the method uses a population of size 1, that is, it moves from a current solution to a new one; “P” indicates that a population of size P is used.

10.6 OTHER SOFTWARE FOR GLOBAL OPTIMIZATION

In addition to the Premium Excel Solver and Optquest, there are many other software systems for constrained global optimization; see Pintér (1996b), Horst and Pardalos (1995), and Pintér (1999) for further information. Perhaps the most widely used of these is LGO (Pintér, 1999), (Pintér, 2000), which is intended for smooth problems with continuous variables. It is available as an interactive development environment with a graphical user interface under Microsoft Windows, or as a callable library, which can be invoked from an application written by the user in

Fortran, C/C++, Visual Basic, or Delphi. The user provides the model coded as a corresponding subroutine or function.

LGO operates in two phases. The first is the global phase, which attempts to find a point which is a good approximation to a global optimum. It uses an adaptive deterministic as well as a random sampling technique, with an option to apply these within a branch-and-bound procedure. The ensuing local phase starts from this point and finds an improved point, which is the “nearest” local optimum, using a combination of local gradient-based NLP algorithms.

REFERENCES

- Avriel, M. *Nonlinear Programming*. Prentice-Hall, Englewood Cliffs, NJ (1976).
- Barnes, J. W.; and L. K. Vanston. “Scheduling Jobs with Linear Delay Penalties and Sequence Dependent Setup Costs.” *Oper Res* **29**: (1) 146–161 (1981).
- Crainic, T. G.; and G. Laporte, eds. *Fleet Management and Logistics*. Kluwer Academic Publishers, Boston/Dordrecht/London (1998).
- Floquet, P.; L. Pibouleau; and S. Domenech. “Separation Sequence Synthesis: How to Use a Simulated Annealing Procedure.” *Comput Chem Eng* **18**: 1141–1148 (1994).
- Floudas, C. A. “Global Optimization in Design and Control of Chemical Process Systems.” *J Process Cont* **10**: 125–134 (2000a).
- Floudas, C. A. *Deterministic Global Optimization: Theory, Methods, and Applications*. Kluwer Academic Publishers, Norwell, MA (2000b).
- Fogel, D. B. “A Comparison of Evolutionary Programming and Genetic Algorithms on Selected Constrained Optimization Problems.” *Simulation* **64**: 397–404 (1995).
- Glover, F. *A Template for Scatter Search and Path Relinking*, working paper, School of Business, University of Colorado, Boulder, CO, 80309 (1998).
- Glover, F.; and M. Laguna. *Tabu Search*. Kluwer Academic Publishers, Norwell, MA (1997).
- Hiriart-Urruty, J. D.; and C. Lemarechal. *Convex Analysis and Minimization Algorithms*. Springer-Verlag, Berlin (1993).
- Holland, J. H. *Adaptations in Natural and Artificial Systems*. University of Michigan Press, Ann Arbor, MI (1975), reissued by MIT Press, Cambridge, MA (1992).
- Horst, R.; and P. M. Pardalos. *Handbook of Global Optimization*. Kluwer Academic Publishers, Dordrecht/Boston/London (1995).
- Johnston, D. S.; C. R. Aragon; L. A. McGeoch; et al. “Optimization by Simulated Annealing: An Experimental Evaluation: Part 1, Graph Partitioning.” *Oper Res* **37**: 865–892 (1989).
- Kearfott, R. B. *Rigorous Global Search: Continuous Problems*. Kluwer Academic Publishers, Norwell, MA (1996).
- Laguna, M.; J. W. Barnes; and F. Glover. “Tabu Search Methods for a Single Machine Scheduling Problem.” *J Intell Manufact* **2**: 63 (1991).
- Locatelli, M.; and F. Schoen. “Random Linkage: A Family of Acceptance/Rejection Algorithms for Global Optimization.” *Math Prog* **85**: 379–396 (1999).
- Pintér, J. D. *Global Optimization in Action (Continuous and Lipschitz Optimization: Algorithms, Implementations, and Applications)*. Kluwer Academic Publishers, Norwell, MA (1996a).

- Pintér, J. D. “Continuous Global Optimization Software: a Brief Review.” *Optima* **52**: 1–8 (1996b).
- Pintér, J. D. “Continuous Global Optimization.” *Interactive Transactions of ORMS* **2** (1999). Available online at <http://catt.bus.okstate.edu/itorms>.
- Pintér, J. D. *Computational Global Optimization in Nonlinear Systems: An Interactive Tutorial*. Published for INFORMS by Lionheart Publishing, Atlanta (2000). Available online at www.lionhrtpub.com/books.
- Reeves, C. R. “Genetic Algorithms for the Operations Researcher.” *INFORMS J Comput* **9**(3): 231–250 (1997).
- Rinnooy Kan, A. H. G.; and G. T. Timmer. “Stochastic Global Optimization Methods, Part 2: Multi Level Methods.” *Math Prog* **39**: 57–78 (1987).
- Rinnooy Kan, A. H. G.; and G. T. Timmer. “Global Optimization,” Chapter 9 In *Handbooks in OR and MS, vol. 1*. G. L. Nemhauser et al., eds. Elsevier Science Publishers B. V., Amsterdam, The Netherlands (1989).

SUPPLEMENTARY REFERENCES

- Adjiman, C. S.; and C. A. Floudas. “Rigorous Convex Underestimators for General Twice-Differentiable Problems.” *J Global Optim* **9**: 23 (1996).
- Adjiman, C. S.; I. P. Androulakis; C. D. Maranas; and C. A. Floudas. “A Global Optimization Method aBB for Process Design.” *Comput Chem Eng* **20**: S419–424 (1996).
- Adjiman, C. S.; I. P. Androulakis; and C. A. Floudas. “A Global Optimization Method aBB, for General Twice-Differentiable Constrained NLPs II. Implementation and Computational Results.” *Comput Chem Eng* **22**: 1159–1179 (1998).
- Adjiman, C. S.; S. Dallwig; C. A. Floudas; and A. Neumaier. “A Global Optimization Method, aBB, for General Twice-Differentiable Constrained NLPs—I, Theoretical Advances.” *Comput Chem Eng* **22**: 1137–1158 (1998).
- Angeline, P. J.; and K. E. Kinneer Jr. *Advances in Genetic Programming, vol. 2*. MIT Press, Cambridge, MA (1998).
- Azzaro-Pantel, C.; L. Bernal-Haro; P. Baudet; S. Demenech, et al. “A Two-Stage Methodology for Short-term Batch Plant Scheduling: Discrete-Event Simulation and Genetic Algorithm.” *Comput Chem Eng* **22**: 1461–1481 (1998).
- Choi, H.; J. W. Ko; and V. Manousiouthakis. “A Stochastic Approach to Global Optimization of Chemical Processes.” *Comput Chem Eng* **23**: 1351–1358 (1999).
- Esposito, W. R.; and C. A. Floudas. “Parameter Estimation in Nonlinear Algebraic Models via Global Optimization.” *Comput Chem Eng* **22**: S213–220 (1998).
- Fogel, D. B. “A Comparison of Evolutionary Programming and Genetic Algorithms on Selected Constrained Optimization Problems.” *Simulation* **64**: 3499 (1995).
- Friese, T.; P. Ulbig; and S. Schulz. “Use of Evolutionary Algorithms for the Calculation of Group Contribution Parameters in Order to Predict Thermodynamic Properties. Part 1: Genetic Algorithms.” *Comput Chem Eng* **22**: 1559–1572 (1998).
- Garrard, A.; and E. S. Fraga. “Mass Exchange Network Synthesis Using Genetic Algorithms.” *Comput Chem Eng* **22**: 1837–1850 (1998).
- Greiff, D. J.; and C. Aldrich. “Empirical Modelling of Chemical Process Systems with Evolutionary Programming.” *Comput Chem Eng* **22**: 995–1005 (1998).
- Gross, B.; and P. Roosen. “Total Process Optimization in Chemical Engineering with Evolutionary Algorithms.” *Comput Chem Eng* **22**: S229–236 (1998).

- Hanagandi, V.; and M. Nikolaou. "A Hybrid Approach to Global Optimization Using a Clustering Algorithm in a Genetic Search Framework." *Comput Chem Eng* **22**: 1913–1925 (1998).
- Haupt, R. L. *Practical Genetic Algorithms*. Wiley, New York (1998).
- Jung, J. H.; C. H. Lee; and I-B. Lee. "A Genetic Algorithm for Scheduling of Multi-Product Batch Processes." *Comput Chem Eng* **22**: 1725–1730 (1998).
- Karr, C. L.; and L. M. Freeman. *Industrial Applications of Genetic Algorithms*. CRC Press, Boca Raton, FL (1998).
- Löhl, T.; C. Schulz; and S. Engell. "Sequencing of Batch Operations for Highly Coupled Production Process: Genetic Algorithms Versus Mathematical Programming." *Comput Chem Eng* **22**: S579–585 (1998).
- Mitchell, M. *An Introduction to Genetic Algorithms*. The MIT Press, Cambridge, MA (1998).
- Pham, Q.T. "Dynamic Optimization of Chemical Engineering Processes by an Evolutionary Method." *Comput Chem Eng* **22**: 1089–1097 (1998).
- Sen, S.; S. Narasimhan; and K. Deb. "Sensor Network Design of Linear Processes Using Genetic Algorithms." *Comput Chem Eng* **22**: 385–390 (1998).
- Wang, K.; Löhl, T.; Stobbe, M.; and S. Engell. "A Genetic Algorithm for Online-scheduling of a Multiproduct Polymer Batch Plant." *Comput Chem Eng* **24**: 393–400 (2000).
- Zamora, J. M.; and I. E. Grossmann. "Continuous Global Optimization of Structured Process Systems Models." *Comput Chem Eng* **22**: 1749–1770 (1998).

PART III

APPLICATIONS OF OPTIMIZATION

THIS SECTION OF the book is devoted to representative applications of the optimization techniques presented in Chapters 4 through 10. Chapters 11 through 16 include the following major application areas:

1. Heat transfer and energy conservation (Chapter 11)
2. Separations (Chapter 12)
3. Fluid flow (Chapter 13)
4. Reactors (Chapter 14)
5. Large-scale plant design and operations (Chapter 15)
6. Integrated planning, scheduling, and control (Chapter 16)

Each chapter presents several detailed studies illustrating the application of various optimization techniques. The following matrix shows the classification of the examples with respect to specific techniques. Truly optimal design of process plants cannot be performed by considering each unit operation separately. Hence, in Chapter 15 we discuss the optimization of large-scale plants, including those represented by flowsheet simulators.

We have not included any homework problems in Chapters 11 through 16. As a general suggestion for classroom use, parameters or assumptions in each example can be changed to develop a modified problem. By changing the numerical method employed or the computer code one can achieve a variety of problems.

Classification of optimization applications (example number is in parentheses) by technique

Methods	Chapter					
	11	12	13	14	15	16
Analytical solution	Waste heat recovery (11.1)		Pipe diameter (13.1)			Material balance reconciliation (16.4)
One-dimensional search	Multistage evaporator (11.3)	Reflux ratio of distillation column (12.4)	Fixed-bed filter (13.3)			
Unconstrained optimization		Nonlinear regression of VLE data (12.3)	Minimum work of compression (13.2)			
Linear programming	Boiler/turbo generator system (11.4)			Thermal cracker (14.1)		Planning and scheduling (16.1)
Nonlinear programming		Staged-Distillation column (12.1) Liquid extraction column (12.2)	Gas transmission network (13.4)	Ammonia reactor (14.2) Alkylation reactor (14.3) CVD reactor (14.5)	Refrigeration process (15.2) Extractive distillation (15.3) Operating margin (15.4)	Reactor control (16.3)
Mixed integer programming	Heat exchanger (11.2)		Gas transmission network (13.4)	Protein folding (14.4) Reaction synthesis (14.6)		Batch scheduling (16.2)

HEAT TRANSFER AND ENERGY
CONSERVATION

Example

11.1 Optimizing Recovery of Waste Heat	419
11.2 Optimal Shell-and-Tube Heat Exchanger Design	422
11.3 Optimization of a Multi-Effect Evaporator	430
11.4 Boiler/Turbo-Generator System Optimization	435
References	438
Supplementary References	439

A VARIETY OF AVAILABLE energy conservation measures can be adopted to optimize energy usage throughout a chemical plant or refinery. The following is a representative list of design or operating factors related to heat transfer and energy use that can involve optimization:

1. Fired heater combustion controls
2. Heat recovery from stack gases
3. Fired heater convection section cleaning
4. Heat exchanger network configuration
5. Extended surface heat exchanger tubing to improve heat transfer
6. Scheduling of heat exchanger cleaning
7. Air cooler performance
8. Fractionating towers: optimal reflux ratio, heat exchange, and so forth
9. Instrumentation for monitoring energy usage
10. Reduced leakage in vacuum systems and pressure lines and condensers
11. Cooling water savings
12. Efficient water treatment for steam raising plants
13. Useful work from steam pressure reduction
14. Steam traps, tracing, and condensate recovery
15. CO boilers on catalytic cracking units
16. Electrical load leveling
17. Power factor improvement
18. Power recovery from gases or liquids
19. Loss control in refineries
20. Catalyst improvements

Many of the conservation measures require detailed process analysis plus optimization. For example, the efficient firing of fuel (category 1) is extremely important in all applications. For any rate of fuel combustion, a theoretical quantity of air (for complete combustion to carbon dioxide and water vapor) exists under which the most efficient combustion occurs. Reduction of the amount of air available leads to incomplete combustion and a rapid decrease in efficiency. In addition, carbon particles may be formed that can lead to accelerated fouling of heater tube surfaces. To allow for small variations in fuel composition and flow rate and in the air flow rates that inevitably occur in industrial practice, it is usually desirable to aim for operation with a small amount of excess air, say 5 to 10 percent, above the theoretical amount for complete combustion. Too much excess air, however, leads to increased sensible heat losses through the stack gas.

In practice, the efficiency of a fired heater is controlled by monitoring the oxygen concentration in the combustion products in addition to the stack gas temperature. Dampers are used to manipulate the air supply. By tying the measuring instruments into a feedback loop with the mechanical equipment, optimization of operations can take place in real time to account for variations in the fuel flow rate or heating value.

As a second example (category 4), a typical plant contains large numbers of heat exchangers used to transfer heat from one process stream to another. It is important to continue to use the heat in the streams efficiently throughout the process. Incoming crude oil is heated against various product and reflux streams

before entering a fired heater in order to be brought to the desired fractionating column flash zone temperature. Among the factors that must be considered in design or retrofit are

1. What should be the configuration of flows (the order of heat exchange for the crude oil)?
2. How much heat exchange surface should be supplied within the chosen configuration?

Additional heat exchange surface area leads to improved heat recovery in the crude oil unit but increases capital costs so that increasing the heat transfer surface area soon reaches diminishing returns. The optimal configuration and areas selected, of course, are strongly dependent on fuel costs. As fuel costs rise, existing plants can usually profit from the installation of additional heat exchanger surface in circumstances previously considered only marginally economic.

As a final example (category 6), although heat exchangers may be very effective when first installed, many such systems become dirty in use and heat transfer rates deteriorate significantly. It is therefore often useful to establish optimal heat exchanger cleaning schedules. Although the schedules can be based on observations of the actual deterioration of the overall heat transfer of the exchanger in question, it is also possible to optimize the details of the cleaning schedules depending on an economic assessment of each exchanger.

In this chapter we illustrate the application of various optimization techniques to heat-transfer-system design. First we show how simple rules of thumb on boiler temperature differences can be derived (Example 11.1). Then a more complicated design of a heat exchanger is examined (Example 11.2), leading to a constrained optimization problem involving some discrete-valued variables. Example 11.3 discusses the use of optimization in the design and operation of evaporators, and we conclude this chapter by demonstrating how linear programming can be employed to optimize a steam/power system (Example 11.4). For optimization of heat exchanger networks by mathematical programming methods, refer to Athier et al. (1997), Briones and Kokossis (1996), and Zamora and Grossmann (1998).

EXAMPLE 11.1 OPTIMIZING RECOVERY OF WASTE HEAT

A variety of sources of heat at elevated temperatures exist in a typical chemical plant that may be economically recoverable for production of power using steam or other working fluids, such as freon or light hydrocarbons. Figure E11.1 is a schematic of such a system. The system power output can be increased by using larger heat exchanger surface areas for both the boiler and the condenser. However, there is a trade-off between power recovery and capital cost of the exchangers. Jegede and Polley (1992), Reppich and Zagermann (1995), Sama (1983), Swearingen and Ferguson (1984), and Steinmeyer (1984) have proposed some simple rules based on analytical optimization of the boiler ΔT .

In a power system, the availability expended by any exchanger is equal to the net work that could have been accomplished by having each stream exchange heat with the surroundings through a reversible heat engine or heat pump. In the boiler in Figure E11.1, heat is transferred at a rate Q (the boiler load) from the average hot fluid

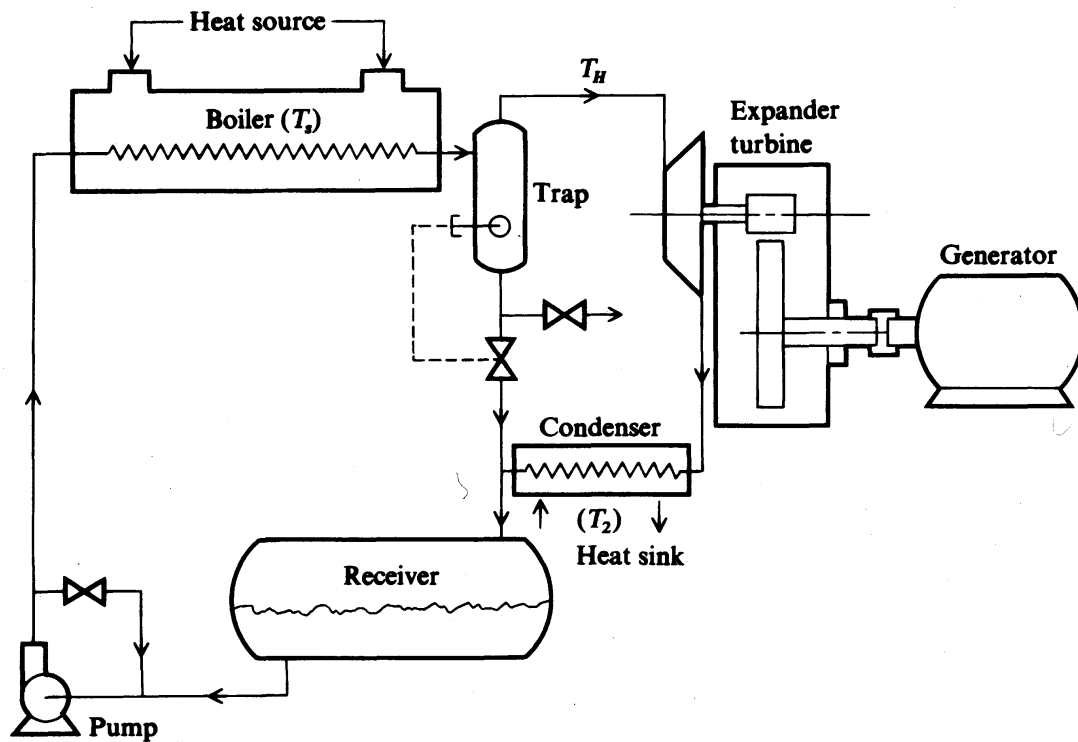


FIGURE E11.1
Schematic of power system.

temperature T_s to the working fluid at T_H . The working fluid then exchanges heat with the condenser at temperature T_2 . If we ignore mechanical friction and heat leaks, the reversible work available from Q at temperature T_s with the condensing (cold-side) temperature at T_2 is

$$W_1 = Q \left(\frac{T_s - T_2}{T_s} \right) \quad (a)$$

The reversible work available from the condenser using the working fluid temperature T_H (average value) and the heat sink temperature T_2 is

$$W_2 = Q \left(\frac{T_H - T_2}{T_H} \right) \quad (b)$$

Hence the ideal power available from the boiler can be found by subtracting W_2 from W_1

$$W_2 - W_1 = \Delta W = Q \left(\frac{T_2}{T_H} - \frac{T_2}{T_s} \right) \quad (c)$$

In this expression T_s and T_2 are normally specified, and T_H is the variable to be adjusted. If Q is expressed in Btu/h, and the operating cost is C_{op} , then the value of the available power is

$$C_{op} = C_H \eta \gamma Q \left(\frac{T_2}{T_H} - \frac{T_2}{T_s} \right) \quad (d)$$

where η = overall system efficiency (0.7 is typical)

y = number of hours per year of operation

C_H amalgamates the value of the power in \$/kWh and the necessary conversion factors to have a consistent set of units

You can see, using Equation (d) only, that C_{op} is minimized by setting $T_H = T_s$ (infinitesimal boiler ΔT). However, this outcome increases the required boiler heat transfer area to an infinite area, as can be noted from the calculation for the area

$$A = \frac{Q}{U(T_s - T_H)} \quad (e)$$

(In Equation (e) an average value for the heat transfer coefficient U is assumed, ignoring the effect of pressure drop. U depends on the working fluid and the operating temperature.) Let the cost per unit area of the exchanger be C_A and the annualization factor for capital investment be denoted by r . Then the annualized capital cost for the boiler is

$$C_c = \frac{C_A Q r}{U(T_s - T_H)} \quad (f)$$

Finally, the objective function to be minimized with respect to T_H , the working fluid temperature, is the sum of the operating cost and surface area costs:

$$f = C_H \eta y Q \left(\frac{T_2}{T_H} - \frac{T_2}{T_s} \right) + \frac{C_A Q r}{U(T_s - T_H)} \quad (g)$$

To get an expression for the minimum of f , we differentiate Equation (g) with respect to T_H and equate the derivative to zero to obtain

$$C_H \eta y Q \left(-\frac{T_2}{T_H^2} \right) + \frac{C_A Q r}{U(T_s - T_H)^2} = 0 \quad (h)$$

To solve the quadratic equation for T_H , let

$$\alpha_1 = C_H \eta y T_2 U$$

$$\alpha_2 = C_A r$$

Q cancels in both terms. On rearrangement, the resulting quadratic equation is

$$(\alpha_1 - \alpha_2) T_H^2 - 2\alpha_1 T_s T_H + \alpha_1 T_s^2 = 0 \quad (i)$$

The solution to (i) for $T_H < T_s$ is

$$T_H = T_s \left(\frac{\alpha_1 - \sqrt{\alpha_1 \alpha_2}}{\alpha_1 - \alpha_2} \right) \quad (j)$$

For a system with $C_A = \$25/\text{ft}^2$, a power cost of $\$0.06/\text{kWh}$ ($C_H = 1.76 \times 10^{-5}$), $U = 95 \text{ Btu}/(\text{h})(^\circ\text{R})(\text{ft}^2)$, $y = 8760 \text{ h/year}$, $r = 0.365$, $\eta = 0.7$, $T_2 = 600^\circ\text{R}$, and $T_s = 790^\circ\text{R}$, the optimal value T_H is 760.7°R , giving a ΔT of 29.3°R . Swearingen and Ferguson showed that Equation (h) can be expressed implicitly as

$$\Delta T = T_s - T_H = T_H \left(\frac{\alpha_1}{\alpha_2} \right)^{1/2} \quad (k)$$

In this form, it appears that the allowable ΔT increases as the working fluid temperature increases. This suggests that the optimum ΔT for a heat source at 900°R is lower than that for a heat source at 1100°R . In fact, Equation (j) indicates that the optimum ΔT is directly proportional to T_s . Sama argues that this is somewhat counterintuitive because the Carnot "value" of a high-temperature source implies using a smaller ΔT to reduce lost work.

The working fluid must be selected based on the heat source temperature, as discussed by Swearingen and Ferguson. See Sama for a discussion of optimal temperature differences for refrigeration systems; use of Equation (k) leads to ΔT 's ranging from 8 to 10°R .

EXAMPLE 11.2 OPTIMAL SHELL-AND-TUBE HEAT EXCHANGER DESIGN

In this example we examine a procedure for optimizing the process design of a baffled shell-and-tube, single-pass, counterflow heat exchanger (see Figure E11.2a), in which the tube fluid is in turbulent flow but no change of phase of fluids takes place in the shell or tubes. Usually the following variables are specified a priori by the designer:

1. Process fluid rate (the hot fluid passes through the tubes), W_i
2. Process fluid temperature change, $T_2 - T_1$

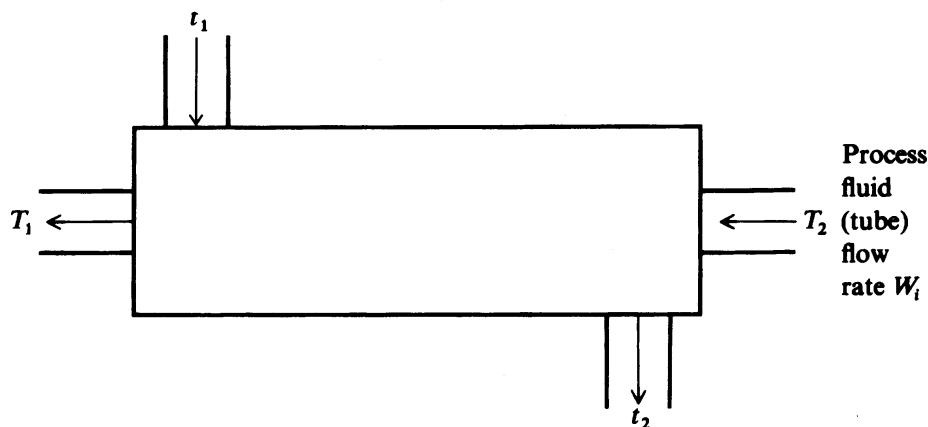


FIGURE E11.2a

Process diagram of shell-and-tube counterflow heat exchanger. Key: $\Delta t_1 = T_1 - t_1$ cold-end temperature difference; $\Delta t_2 = T_2 - t_2$ warm-end temperature difference.

3. Coolant inlet temperature (the coolant flows through the shell), t_1
4. Tube spacing and tube inside and outside diameters (D_i , D_o).

Conditions 1 and 2 imply the heat duty Q of the exchanger is known.

The variables that might be calculated via optimization include

1. Total heat transfer area, A_o
2. Warm-end temperature approach, Δt_2
3. Number and length of tubes, N_t and L
4. Number of baffle spacings, n_b
5. Tube-side and shell-side pressure drop
6. Coolant flow, W_c

Not all of these variables are independent, as shown in the following discussion.

In contrast to the analysis outlined in Example 11.1, the objective function in this example does not make use of reversible work. Rather, a cost is assigned to the usage of coolant as well as to power losses because of the pressure drops of each fluid. In addition, annualized capital cost terms are included. The objective function in dollars per year is formulated using the notation in Table E11.2A

$$C = C_c W_c y + C_A A_o + C_i E_i A_o + C_o E_o A_o \quad (a)$$

Suppose we minimize the objective function using the following set of four variables, a set slightly different from the preceding list.

1. Δt_2 : warm-end temperature difference
2. A_o : tube outside area
3. h_i : tube inside heat transfer coefficient
4. h_o : tube outside heat transfer coefficient

Only three of the four variables are independent. If A_o , h_i , and h_o are known, then Δt_2 can be found from the heat duty of the exchanger Q :

$$Q = F_t U_o A_o \frac{\Delta t_2 - \Delta t_1}{\ln(\Delta t_2 / \Delta t_1)} \quad (b)$$

F_t is unity for a single-pass exchanger. U_o is given by the values of h_o , h_i , and the fouling coefficient h_f as follows:

$$\frac{1}{U_o} = \frac{1}{f_A h_i} + \frac{1}{h_o} + \frac{1}{h_f} \quad (c)$$

Cichelli and Brinn (1956) showed that the annual pumping loss terms in Equation (a) could be related to h_i and h_o by using friction factor and j -factor relationships for tube flow and shell flow:

$$E_i = \phi_i h_i^{3.5} \quad (d)$$

$$E_o = \phi_o h_o^{4.75} \quad (e)$$

TABLE E11.2A
Nomenclature for heat exchanger optimization

A_{lm}	Log mean of inside and outside tube surface areas
A_i	Inside tube surface area, ft ²
A_o	Outside tube surface area, ft ²
C	Total annual cost, \$/year
C_A	Annual cost of heat exchanger per unit outside tube surface area, \$(/ft ²)(year)
C_c	Cost of coolant, \$/lb mass
C_i	Annual cost of supplying 1(ft)(lb _f)/h to pump fluid flowing inside tubes, \$(h)/(ft)(lb _f)(year)
C_o	Annual cost of supplying 1(ft)(lb _f)/h to pump shell side fluid, \$(h)/(ft)(lb _f)(year)
c	Specific heat at constant pressure, Btu/(lb _m)(°F)
D_i	Tube inside diameter, ft
D_o	Tube outside diameter, ft
E_i	Power loss inside tubes per unit outside tube area, (ft)(lb _f)/(ft ²)(h)
E_o	Power loss outside tubes per unit outside tube area, (ft)(lb _f)/(ft ²)(h)
f	Friction factor, dimensionless
f_A	A_i/A_o
F_t	Multipass exchanger factor
g_c	Conversion factor, (ft)(lb _m)/(lb _f)(h ²) = 4.18 × 10 ⁸
h_f	Fouling coefficient
h_i	Coefficient of heat transfer inside tubes, Btu/(h)(ft ²)(°F)
h_o	Coefficient of heat transfer outside tubes, Btu/(h)(ft ²)(°F)
h_t	Combined coefficient for tube wall and dirt films, based on tube outside area Btu/(h)(ft ²)(°F)
$\frac{1}{h_t} = \frac{L'A_o}{k_w A_{lm}} + \frac{1}{h_f} \frac{A_o}{A_i} + \frac{1}{h_o}$	
k	Thermal conductivity, Btu/(h)(ft)(°F)
L	Lagrangian function
L_t	Length of tubes, ft
L'	Thickness of tube wall, ft

(continued)

The coefficients ϕ_i and ϕ_o depend on fluid specific heat c , thermal conductivity k , density ρ , and viscosity μ , as well as the tube diameters. ϕ_o is based on either in-line or staggered tube arrangements.

If we solve for W_c from the energy balance

$$W_c = \frac{Q}{c(\Delta t_1 - \Delta t_2 + T_2 - T_1)} \quad (f)$$

and substitute for E_i , E_o , and W_c in Equation (a), the resulting objective function is

$$f = \frac{C_c y Q}{c(\Delta t_1 - \Delta t_2 + T_2 - T_1)} + C_A A_o + C_i \phi_i h_i^{3.5} A_o + C_o \phi_o h_o^{4.75} A_o \quad (g)$$

TABLE E11.2A (CONTINUED)
Nomenclature for heat exchanger optimization

n_b	Number of baffle spacing on shell side = number of baffles plus 1
N_c	Number of clearances for flow between tubes across shell axis
N_t	Number of tubes in exchanger
Δp_i	Pressure drop for flow through tube side, lb _f /ft ²
Δp_o	Pressure drop for flow through tube side, lb _f /ft ²
Q	Heat transfer rate in heat exchanger, Btu/h
S_0	Minimum cross-sectional area for flow across tubes, ft ²
T_1	Outlet temperature of process fluid, °F
T_2	Inlet temperature of process fluid, °F
t_1	Inlet temperature of coolant, °F
t_2	Outlet temperature of coolant, °F
ΔT_1	$T_1 - t_1$, = cold-end temperature difference
ΔT_2	$T_2 - t_2$, = warm-end temperature difference
U_o	Overall coefficient of heat transfer, based on outside tube area, Btu/(h)(ft ²)(°F)
v_i	Average velocity of fluid inside tubes, ft/h
v_o	Average velocity of fluid outside tubes, ft/h at shell axis
W_c	Coolant rate, lb/h
W_i	Flow rate of fluid inside tubes, lb _m /h
W_o	Flow rate of fluid outside tubes, lb _m /h
y	Operating hours per year
ρ_i	Density of fluid inside tubes, lb _m /ft ³
ρ_o	Density of fluid outside tubes, lb _m /ft ³
μ	Viscosity of fluid, lb _m /(h)(ft)
ϕ_i	Factor relating friction loss to h_i
ϕ_o	Factor relating friction loss to h_o
ω	Lagrange multiplier

Subscripts

c	Coolant
f	Film temperature, midway between bulk fluid and wall temperature
i	Inside the tubes
o	Outside the tubes
w	Wall

To accommodate the constraint (b), a Lagrangian function L is formed by augmenting f with Equation (b), using a Lagrange multiplier ω

$$L = f + \omega \left[\frac{F_t(\Delta t_2 - \Delta t_1)}{Q \ln(\Delta t_2/\Delta t_1)} - \frac{1}{U_o A_o} \right] \quad (h)$$

Equation (h) can be differentiated with respect to four variables (h_i , h_o , Δt_2 , and A_o). After some rearrangement, you can obtain a relationship between the optimum h_o and h_i , namely

$$h_o = \left(\frac{0.74 C_i \phi_i f_A}{C_o \phi_o} \right)^{0.17} h_i^{0.78} \quad (i)$$

This is the same result as derived by McAdams (1942), having the interpretation that the friction losses in the shell and tube sides, and the heat transfer resistances must be balanced economically. The value of h_i can be obtained by solving

$$C_A - 2.5C_i\phi_i h_i^{3.5} - 2.91(C_o\phi_o)^{0.17}(C_i\phi_i f_A)^{0.83} h_i^{3.72} - \frac{3.5C_i\phi_i f_A h_i^{4.5}}{h_i} = 0 \quad (j)$$

The simultaneous solution of Equations (f), (i), and (j) yields another expression:

$$\frac{C_c y U_o}{c(C_A + C_i E_i + C_o E_o)} = \left(1 + \frac{T_2 - T_1}{\Delta t_2 - \Delta t_1}\right)^2 \left[\ln\left(\frac{\Delta t_2}{\Delta t_1}\right) - 1 + \frac{\Delta t_2}{\Delta t_1} \right] \quad (k)$$

The following algorithm can be used to obtain the optimal values of h_i , h_o , A_o , and Δt_2 without the explicit calculation of ω :

1. Solve for h_i from Equation (j)
2. Obtain h_o from Equation (i)
3. Calculate U_o from Equation (c)
4. Determine E_i and E_o from h_i and h_o using Equations (d) and (e) and obtain Δt_2 by solving Equation (k)
5. Calculate A_o from Equation (b)
6. Find W_c from Equation (f)

Note that steps 1 to 6 require that several nonlinear equations be solved one at a time. Once these variables are known, the physical dimensions of the heat exchanger can be determined.

7. Determine the optimal v_i and v_o from h_i and h_o using the appropriate heat transfer correlations (see McAdams, 1942); recall that the inside and outside tube diameters are specified a priori.
8. The number of tubes N_t can be found from a mass balance:

$$v_i N_t \frac{\pi D_i^2}{4} = W_i \quad (l)$$

9. The length of the tubes L_t can be found from

$$A_o = N_t \pi D_o L_t \quad (m)$$

10. The number of clearances N_c can be found from N_p , based on either square pitch or equilateral pitch. The flow area S_o is obtained from v_o (flow normal to a tube bundle). Finally, baffle spacing (or the number of baffles) is computed from S_o , A_o , N_p , and N_c .

Having presented the pertinent equations and the procedure for computing the optimum, let us check the approach by computing the degrees of freedom in the design problem.

Design Variables

W_i , T_1 , T_2 , t_1 , tube spacing, D_i , D_o , Q
 Δt_2 , W_c , A_o , N_t , L_t , U_o , n_b , Δp_p , Δp_s , v_i , v_o , h_i , h_o

Status

(number of variables)

Given (8)

Unspecified (13)

Total number of variables = 8 + 13 = 21

<i>Design Relationships</i>	<i>Number of Equations</i>
1. Equations (b), (c), (d), (e) (f), (l), (m)	7
2. Heat transfer correlations for h_i and h_o (step 7)	2
3. $W_c = \rho_o v_o s_o$ (step 10)	1
Total number of relationships	10

Degrees of freedom for optimization = total number of variables – number of given variables – number of equations

$$= 21 - 8 - 10 = 3$$

Note this result agrees with Equation (h) in that four variables are included in the Lagrangian, but with one constraint corresponding to 3 degrees of freedom.

Several simplified cases may be encountered in heat exchanger design.

Case 1. U_o is specified and pressure drop costs are ignored in the objective function. In this case C_i and C_o can be set equal to zero and Equation (k) can be solved for Δt_2 (see Peters and Timmerhaus (1980) for a similar equation for a condensing vapor). Figure E11.2b shows a solution to Equation (k) (Cichelli and Brinn).

Case 2. Coolant flow rate is fixed. Here Δt_2 is known, so the tube side and shell side coefficients and area are optimized. Use Equation (i) and (j) to find h_o and h_i . A_o is then found from Equation (b).

In the preceding analysis no inequality constraints were introduced. As a practical matter the following inequality constraints may apply:

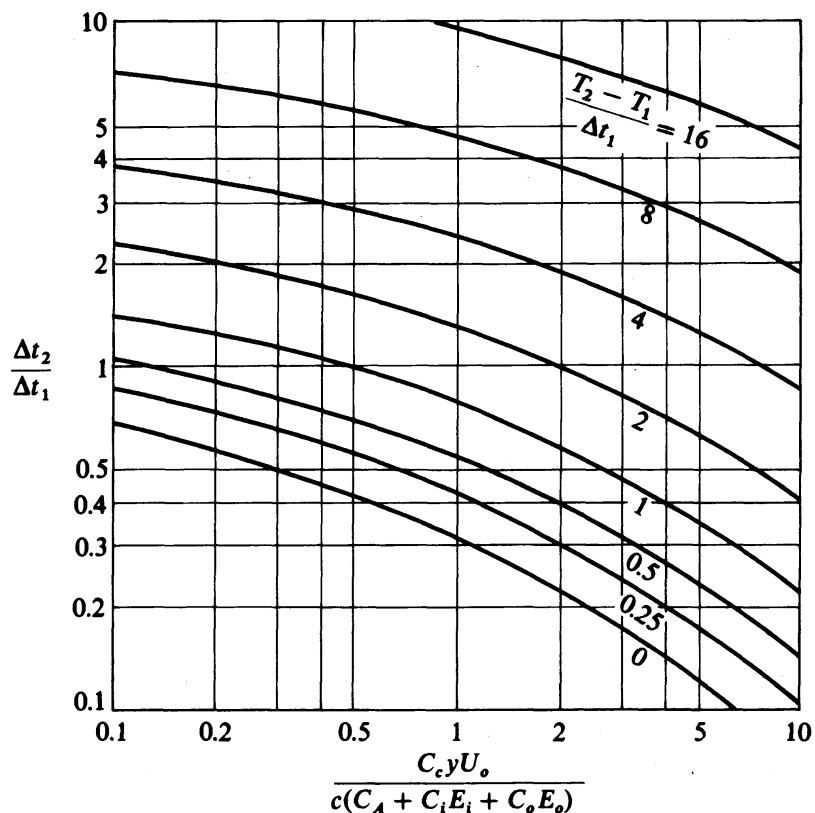


FIGURE E11.2b

Solution to Equation (k) for the case in which U_o is specified and pressure drop costs are ignored.

TABLE E11.2B
Design specifications for one case of heat exchanger optimization

Variables	
Process fluid	Gas
Inlet temperature of process fluid, °F	150
Outlet temperature of process fluid, °F	100
Process fluid flow rate, lb/h	20,000
Maximum process fluid velocity, ft/s	160
Minimum process fluid velocity, ft/s	0.001
Utility fluid	Water
Inlet utility fluid temperature, °F	70
Maximum allowable utility fluid temperature, °F	140
Maximum utility fluid velocity, ft/s	8
Minimum utility fluid velocity, ft/s	0.5
Shell side fouling factor	2000
Tube side fouling factor	1500
Cost of pumping process fluid, \$/(ft)(lb _p)	0.7533×10^{-8}
Cost of pumping utility fluid, \$/(ft)(lb _p)	0.7533×10^{-8}
Cost of utility fluid, \$/lb _m	0.5000×10^{-5}
Factor for pressure	1.45
Cost index	1.22
Fractional annual fixed charges	0.20
Fractional cost of installation	0.15
Tube material	Steel
Type of tube layout	Triangular
Construction type	Fixed tube sheet
Maximum allowable shell diameter, in.	40
Bypassing safety factor	1.3
Constant for evaluating outside film coat	0.33
Hours operation per year	7000
Thermal conductivity of metal Btu/(h)(ft ²)(°F)	26
Number of tube passes	1

Source: Tarrer et al. (1971).

1. Maximum velocity on shell or tube side
2. Longest practical tube length
3. Closest practical baffle spacing
4. Maximum allowable pressure drops (shell or tube side)

The velocity on the tube side can be modified by changing the single-pass design to a multiple-pass configuration. In this case $F_t \neq 1$ in Equation (b). From formulas in McCabe, F_t depends on t_2 (or Δt_2), hence the necessary conditions derived previously would have to be changed. The fluids could be switched (shell vs. tube side) if constraints are violated, but there may well be practical limitations such as one fluid being quite dirty or corrosive so that the fluid must flow in the tube side (to facilitate cleaning or to reduce alloy costs).

Other practical features that must be taken into account are the fixed and integer lengths of tubes (8, 12, 16, and 20 feet), and the maximum pressure drops allowed.

TABLE 11.2C
Optimal solution for a heat exchanger involving discrete variables

Variables	Continuous- Variable Optimal Design	Standard integer sizes			
		1	2	3	4
Tube length, ft	10.5	8	8	12	12
Number of tubes	66	110	85	64	42
Total area, ft ²	193.3	230	178	201	132
Total cost, \$/year	734	908	923	738	784
Heat transfer coefficients, Btu/(h)(ft ²)(°F)					
Outside	554	561	649	512	617
Inside	56.2	37.1	45.9	57.4	80.5
Overall	41.0	28.4	34.5	41.5	56.2
Outlet utility fluid temperature (°F)	117.1	102.1	96.5	120.1	112.4
Utility fluid flow rate, lb _m /h	5306	7790	9422	4993	5897
Inside pressure drop, psi	0.279	0.086	0.138	0.318	0.701
Outside pressure drop, psi	6.45	5.24	7.91	4.98	9.13
Number of baffle spaces	119	85	79	121	119
Shell diameter, in.	12	16	14	12	10

Tube layout: 1.00-in. outside diameter
0.834-in. inside diameter
0.25-in. clearance
0.083-in. wall thickness
1.25-in. pitch

Source: Tarrer et al. (1971).

Although a 20-psi drop may be typical for liquids such as water, higher values are employed for more viscous fluids. Exchanging shell sides with tube sides may mitigate pressure drop restrictions. The tube's outside diameter is specified a priori in the optimization procedure described earlier; usually $\frac{3}{4}$ - or 1-inch outside diameter (o.d.) tubes are used because of their greater availability and ease of cleaning. Limits on operating variables, such as maximum exit temperature of the coolant, maximum and minimum velocities for both streams, and maximum allowable shell area must be included in the problem specifications along with the number of tube passes.

Table 11.2B lists the specifications for a typical exchanger, and Table 11.2C gives the results of optimization for several cases for two standard tube lengths, 8 and 12 ft. The minimum cost occurs for a 12-ft tube length with 64 tubes (case 3). Many commercial codes exist to carry out heat exchanger design. Search the Web for the most recent versions.

EXAMPLE 11.3 OPTIMIZATION OF A MULTI-EFFECT EVAPORATOR

When a process requires an evaporation step, the problem of evaporator design needs serious examination. Although the subject of evaporation and the equipment to carry out evaporation have been studied and analyzed for many years, each application has to receive individual attention. No evaporation configuration and its equipment can be picked from a stock list and be expected to produce trouble-free operation.

An engineer working on the selection of optimal evaporation equipment must list what is “known,” “unknown,” and “to be determined.” Such analysis should at least include the following:

Known

- Production rate and analysis of product
- Feed flow rate, feed analysis, feed temperature
- Available utilities (steam, water, gas, etc.)
- Disposition of condensate (location) and its purity
- Probable materials of construction

Unknown

- Pressures, temperatures, solids, compositions, capacities, and concentrations
- Number of evaporator effects
- Amount of vapor leaving the last effect
- Heat transfer surface

Features to be determined

- Best type of evaporator body and heater arrangement
- Filtering characteristics of any solids or crystals
- Equipment dimensions, arrangement
- Separator elements for purity of overhead vapors
- Materials, fabrication details, instrumentation

Utility consumption

- Steam
- Electric power
- Water
- Air

In multiple-effect evaporation, as shown in Figure E11.3a, the total capacity of the system of evaporation is no greater than that of a single-effect evaporator having a heating surface equal to one effect and operating under the same terminal conditions. The amount of water vaporized per unit surface area in n effects is roughly $1/n$ that of a single effect. Furthermore, the boiling point elevation causes a loss of available temperature drop in every effect, thus reducing capacity. Why, then, are multiple effects often economic? It is because the cost of an evaporator per square foot of surface area decreases with total area (and asymptotically becomes a constant value) so that to achieve a given production, the cost of heat exchange surface can be balanced with the steam costs.

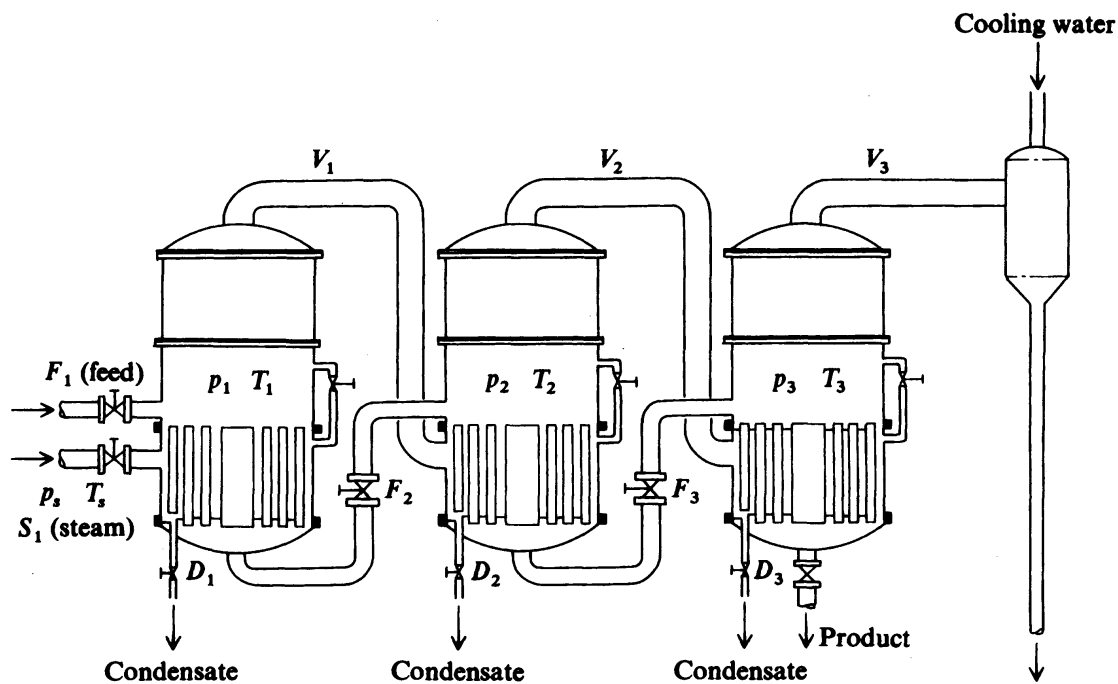


FIGURE E11.3a

Multiple-effect evaporator with forward feed.

Steady-state mathematical models of single- and multiple-effect evaporators involving material and energy balances can be found in McCabe et al. (1993), Yanniotis and Pilavachi (1996), and Esplugas and Mata (1983). The classical simplified optimization problem for evaporators (Schweyer, 1955) is to determine the most suitable number of effects given (1) an analytical expression for the fixed costs in terms of the number of effects n , and (2) the steam (variable) costs also in terms of n . Analytic differentiation yields an analytical solution for the optimal n^* , as shown here.

Assume we are concentrating an inorganic salt in the range of 0.1 to 1.0 wt% using a plant capacity of 0.1–10 million gallons/day. Initially we treat the number of stages n as a continuous variable. Figure E11.3b shows a single effect in the process.

Prior to discussions of the capital and operating costs, we need to define the temperature driving force for heat transfer. Examine the notation in Figure E11.3c; by definition the log mean temperature difference ΔT_{lm} is

$$\Delta T_{lm} = \frac{T_i - T_d}{\ln(T_i/T_d)} \quad (a)$$

Let T_i be equal to constant K for a constant performance ratio P . Because $T_d = T_i - \Delta T_f/n$

$$\Delta T_{lm} = \frac{\Delta T_f/n}{\ln[K/K - (T_f/n)]} \quad (b)$$

Let A = condenser heat transfer areas, ft^2

c_p = liquid heat capacity, $1.05 \text{ Btu}/(\text{lb}_m)(^\circ\text{F})$

C_C = cost per unit area of condenser, $\$6.25/\text{ft}^2$

C_E = cost per evaporator (including partitions), $\$7000/\text{stage}$

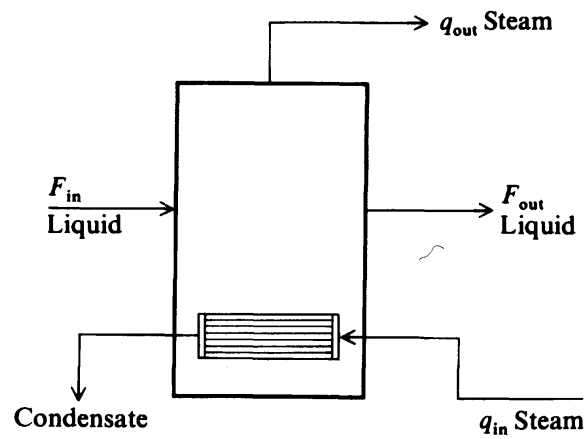


FIGURE E11.3b

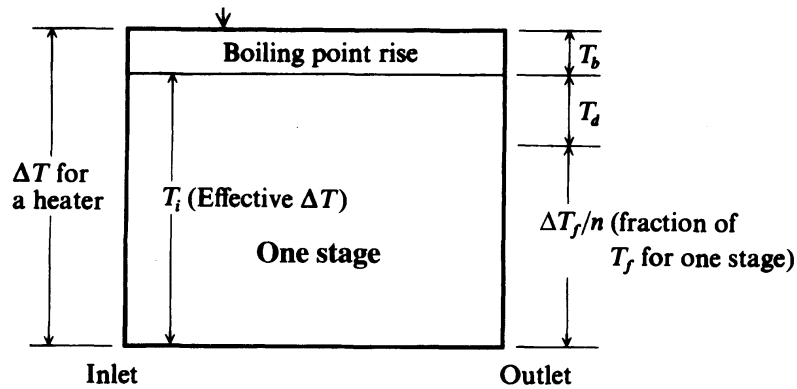


FIGURE E11.3c

- C_S = cost of steam, \$/lb at the brine heater (first stage)
 F_{out} = liquid flow out of evaporator, lb/h
 $K = T_i$, a constant ($T_i = \Delta T - T_b$ at inlet)
 n = number of stages
 P = performance ratio, lb of H_2O evaporated/Btu supplied to brine heater
 Q = heat duty, 9.5×10^8 Btu/h (a constant)
 q_e = total lb H_2O evaporated/h
 q_r = total lb steam used/h
 r = capital recovery factor
 S = lb steam supplied/h
 T_b = boiling point rise, $4.3^\circ F$
 ΔT_f = flash down range, $250^\circ F$
 U = overall heat transfer coefficient (assumed to be constant), $625 \text{ Btu}/(\text{ft}^2)(\text{h})(^\circ F)$
 ΔH_{vap} = heat of vaporization of water, about 1000 Btu/lb

The optimum number of stages is n^* . For a constant performance ratio the total cost of the evaporator is

$$f_1 = C_E n + C_C A \quad (c)$$

For A we introduce

$$A = \frac{Q}{U(\Delta T_{lm})}$$

Then we differentiate f_1 in Equation (c) with respect to n and set the resulting expression equal to zero (Q and U are constant):

$$C_E + C_C \frac{Q}{U} \left[\frac{\partial(1/\Delta T_{lm})}{\partial n} \right]_P = 0 \quad (d)$$

With the use of Equation (b)

$$\left[\frac{\partial(1/\Delta T_{lm})}{\partial n} \right]_P = - \frac{1}{nK(1 - \Delta T_f/nK)} - \frac{\ln(1 - \Delta T_f)}{\Delta T_f} \quad (e)$$

Substituting Equation (e) into (d) plus introducing the values of Q , U , ΔT_f , C_E , and C_C , we get

$$7000 - \left[\frac{(6.25)(9.5 \times 10^8)}{625} \right] \left[\frac{1}{nK(1 - \Delta T_f/nK)} + \frac{\ln(1 - \Delta T_f/nK)}{\Delta T_f} \right] = 0$$

Rearranging

$$\frac{(625)(7000)(250)}{(6.25)(9.5 \times 10^8)} = 0.184 = \frac{250}{nK - 250} + \ln \left(1 - \frac{250}{nK} \right) \quad (f)$$

In practice, as the evaporation plant size changes (for constant Q), the ratio of the stage condenser area cost to the unit evaporator cost remains essentially constant so that the number 0.184 is treated as a constant for all practical purposes. Equation (f) can be solved for nK for constant P

$$nK = 590 \quad (g)$$

Next, we eliminate K from Equation (g) by replacing K with a function of P so that n becomes a function of P . The performance ratio (with constant liquid heat capacity at 347°F) is defined as

$$P = \frac{(\Delta H_{\text{vap}})(q_e)}{(F_{\text{out}} c_{pF} \Delta T_{\text{heater}})_{\text{first stage}}} = \frac{1000}{1.05(4.3 + K)} \frac{q_e}{F_{\text{out}}} \quad (h)$$

The ratio q_e/F can be calculated from

$$\frac{q_e}{F_{\text{out}}} = 1 - \left(\frac{1194 - 322}{1194 - 70} \right)^{1.49} = 0.31$$

where ΔH_{vap} (355°F, 143 psi) = 1194 Btu/lb
 $\Delta H_{\text{liq H}_2\text{O}}$ (350°F) = 322 Btu/lb
 $\Delta H_{\text{liq H}_2\text{O}}$ (100°F) = 70 Btu/lb

Equations (g) and (h) can be solved together to eliminate K and obtain the desired relation

$$\frac{300}{P} - 4.3 = \frac{590}{n^*} \quad (i)$$

Equation (i) shows how the boiling point rise ($T_b = 4.3^\circ\text{F}$) and the number of stages affects the performance ratio.

Optimal performance ratio

The optimal plant operation can be determined by minimizing the total cost function, including steam costs, with respect to P (liquid pumping costs are negligible)

$$f_2 = [C_c A + C_E n]r + C_s S \quad (j)$$

$$rC_c \frac{\partial A}{\partial P} + rC_E \frac{\partial n}{\partial P} + C_s \frac{\partial S}{\partial P} = 0 \quad (k)$$

The quantity for $\partial A/\partial P$ can be calculated by using the equations already developed and can be expressed in terms of a ratio of polynomials in P such as

$$\frac{a(1 + 1/P)}{(1 - bP)^2}$$

where a and b are determined by fitting experimental data. The relation for $\partial n/\partial P$ can be determined from Equation (i). The relation for $\partial S/\partial P$ can be obtained from equation (l)

$$P = \frac{q_e}{Q} = \frac{q_e}{(\Delta H_{\text{vap}})S} = \frac{q_e}{1000S}$$

or

$$S\left(\frac{\text{lb}}{\text{h}}\right) = \frac{q_e}{1000P}$$

or

$$S(\text{lb}) = \frac{\alpha(8760)q_e}{1000P} \quad (l)$$

where α is the fraction of hours per year (8760) during which the system operates.

Equation (k), given the costs, cannot be explicitly solved for P^* , but P^* can be obtained by any effective root-finding technique.

If a more complex mathematical model is employed to represent the evaporation process, you must shift from analytic to numerical methods. The material and enthalpy balances become complicated functions of temperature (and pressure). Usually all of the system parameters are specified except for the heat transfer areas in each effect (n unknown variables) and the vapor temperatures in each effect excluding the last one ($n - 1$ unknown variables). The model introduces n independent equations that serve as constraints, many of which are nonlinear, plus nonlinear relations among the temperatures, concentrations, and physical properties such as the enthalpy and the heat transfer coefficient.

Because the number of evaporators represents an integer-valued variable, and because many engineers use tables and graphs as well as equations for evaporator calculations, some of the methods outlined in Chapters 9 and 10 can be applied for the optimization of multi-effect evaporator cascades.

EXAMPLE 11.4 BOILER/TURBO-GENERATOR SYSTEM OPTIMIZATION

Linear programming is often used in the design and operation of steam systems in the chemical industry. Figure E11.4 shows a steam and power system for a small power house fired by wood pulp. To produce electric power, this system contains two turbo-generators whose characteristics are listed in Table E11.4A. Turbine 1 is a double-extraction turbine with two intermediate streams leaving at 195 and 62 psi; the final stage produces condensate that is used as boiler feed water. Turbine 2 is a single-

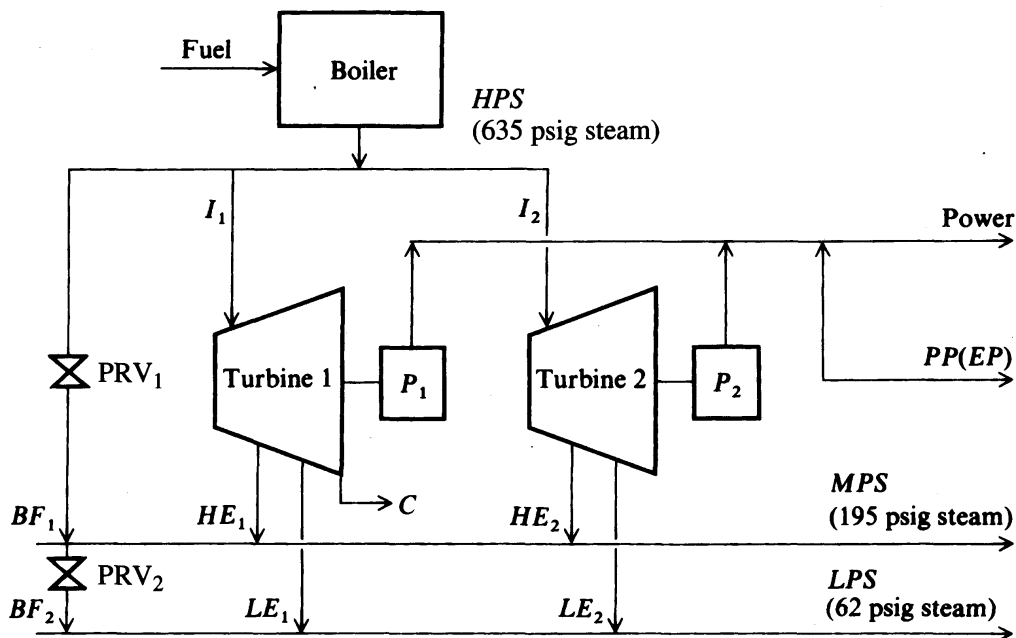


FIGURE E11.4

Boiler/turbo-generator system.

Key: I_i = inlet flow rate for turbine i [lb_m/h]

HE_i = exit flow rate from turbine i to 195 psi header [lb_m/h]

LE_i = exit flow rate from turbine i to 62 psi header [lb_m/h]

C = condensate flow rate from turbine 1 [lb_m/h]

P_i = power generated by turbine i [kW]

BF_1 = bypass flow rate from 635 psi to 195 psi header [lb_m/h]

BF_2 = bypass flow rate from 195 psi to 62 psi header [lb_m/h]

HPS = flow rate through 635 psi header [lb_m/h]

MPS = flow rate through 195 psi header [lb_m/h]

LPS = flow rate through 62 psi header [lb_m/h]

PP = purchased power [kW]

EP = excess power [kW] (difference of purchased power from base power)

PRV = pressure-reducing valve

extraction turbine with one intermediate stream at 195 psi and an exit stream leaving at 62 psi with no condensate being formed. The first turbine is more efficient due to the energy released from the condensation of steam, but it cannot produce as much power as the second turbine. Excess steam may bypass the turbines to the two levels of steam through pressure-reducing valves.

Table E11.4B lists information about the different levels of steam, and Table E11.4C gives the demands on the system. To meet the electric power demand, electric power may be purchased from another producer with a minimum base of 12,000 kW. If the electric power required to meet the system demand is less than this base, the power that is not used will be charged at a penalty cost. Table E11.4D gives the costs of fuel for the boiler and additional electric power to operate the utility system.

The system shown in Figure E11.4 may be modeled as linear constraints and combined with a linear objective function. The objective is to minimize the operating cost of the system by choice of steam flow rates and power generated or purchased, subject to the demands and restrictions on the system. The following objective function is the cost to operate the system per hour, namely, the sum of steam produced HPS , purchased power required PP , and excess power EP :

TABLE 11.4A
Turbine data

Turbine 1		Turbine 2	
Maximum generative capacity	6,250 kW	Maximum generative capacity	9,000 kW
Minimum load	2,500 kW	Minimum load	3,000 kW
Maximum inlet flow	192,000 lb _m /h	Maximum inlet flow	244,000 lb _m /h
Maximum condensate flow	62,000 lb _m /h	Maximum 62 psi exhaust	142,000 lb _m /h
Maximum internal flow	132,000 lb _m /h	High-pressure extraction at	195 psig
High-pressure extraction at	195 psig	Low-pressure extraction at	62 psig
Low-pressure extraction at	62 psig		

TABLE 11.4B
Steam header data

Header	Pressure (psig)	Temperature (°F)	Enthalpy (Btu/lb _m)
High-pressure steam	635	720	1359.8
Medium-pressure steam	195	130 superheat	1267.8
Low-pressure steam	62	130 superheat	1251.4
Feedwater (condensate)			193.0

TABLE 11.4C
Demands on the system

Resource	Demand
Medium-pressure steam (195 psig)	271,536 lb _m /h
Low-pressure steam (62 psig)	100,623 lb _m /h
Electric power	24,550 kW

TABLE 11.4D
Energy data

Fuel cost	\$1.68/10 ⁶ Btu
Boiler efficiency	0.75
Steam cost (635 psi)	\$2.24/10 ⁶ Btu = \$2.24 (1359.8 - 193)/10 ⁶ = \$0.002614/lb _m
Purchased electric power	\$0.0239/kWh average
Demand penalty	\$0.009825/kWh
Base-purchased power	12,000 kW

$$\text{Minimize: } f = 0.00261 HPS + 0.0239 PP + 0.00983 EP \quad (a)$$

The constraints are gathered into the following specific subsets:

Turbine 1

$$\begin{aligned} P_1 &\leq 6250 \\ P_1 &\geq 2500 \\ HE_1 &\leq 192,000 \\ C &\leq 62,000 \\ I_1 - HE_1 &\leq 132,000 \end{aligned} \quad (b)$$

Turbine 2

$$\begin{aligned} P_2 &\leq 9000 \\ P_2 &\geq 3000 \\ I_2 &\leq 244,000 \\ LE_2 &\leq 142,000 \end{aligned} \quad (c)$$

Material balances

$$\begin{aligned} HPS - I_1 - I_2 - BF_1 &= 0 \\ I_1 + I_2 + BF_1 - C - MPS - LPS &= 0 \\ I_1 - HE_1 - LE_1 - C &= 0 \\ I_2 - HE_2 - LE_2 &= 0 \\ HE_1 + HE_2 + BF_1 - BF_2 - MPS &= 0 \\ LE_1 + LE_2 + BF_2 - LPS &= 0 \end{aligned} \quad (d)$$

Power purchased

$$EP + PP \geq 12,000 \quad (e)$$

Demands

$$MPS \geq 271,536$$

$$LPS \geq 100,623 \quad (f)$$

$$P_1 + P_2 + PP \geq 24,550$$

Energy balances

$$1359.8I_1 - 1267.8HE_1 - 1251.4LE_1 - 192C - 3413P_1 = 0 \quad (g)$$

$$1359.8 I_2 - 1267.8 I_2 - 1251.4 LE_2 - 3413 P_2 = 0$$

TABLE E11.4E
Optimal solution to steam system LP

Variable	Name	Value	Status
1	I_1	136,329	BASIC
2	I_2	244,000	BOUND
3	HE_1	128,158	BASIC
4	HE_2	143,377	BASIC
5	LE_1	0	ZERO
6	LE_2	100,623	BASIC
7	C	8,170	BASIC
8	BF_1	0	ZERO
9	BF_2	0	ZERO
10	HPS	380,329	BASIC
11	MPS	271,536	BASIC
12	LPS	100,623	BASIC
13	P_1	6,250	BOUND
14	P_2	7,061	BASIC
15	PP	11,239	BASIC
16	EP	761	BASIC

Value of objective function = 1268.75 \$/h

BASIC = basic variable

ZERO = 0

BOUND = variable at its upper bound

Table E11.4E lists the optimal solution to the linear program posed by Equations (a)–(g). Basic and nonbasic (zero) variables are identified in the table; the minimum cost is \$1268.75/h. Note that $EP + PP$ must sum to 12,000 kWh; in this case the excess power is reduced to 761 kWh.

REFERENCES

Athier, G.; P. Floquet; L. Pibouleau; et al. "Process Optimization by Simulated Annealing and NLP Procedures. Application to Heat Exchanger Network Synthesis." *Comput Chem Eng* **21** (Suppl): S475–S480 (1997).

- Briones, V.; and A. Kokossis. "A New Approach for the Optimal Retrofit of Heat Exchanger Networks." *Comput Chem Eng* **20 (Suppl)**: S43–S48 (1996).
- Cichelli, M. T.; and M. S. Brinn. "How to Design the Optimum Heat Exchanger." *Chem Eng* **196**: May (1956).
- Esplugas, S.; and J. Mata. "Calculator Design of Multistage Evaporators." *Chem Eng* **59** Feb. 7: (1983).
- Jegade, F. O.; and G. T. Polley. "Capital Cost Targets for Networks with Non-Uniform Heat Transfer Specifications." *Comput Chem Eng* **16**: 477 (1992).
- McAdams, W. H. *Heat Transmission*. McGraw-Hill, New York (1942).
- McCabe, W. L.; J. Smith; and P. Harriott. *Unit Operations in Chemical Engineering*, 5th ed. McGraw-Hill, New York (1993).
- Peters, M.; and K. Timmerhaus. *Plant Design and Economics for Chemical Engineers*, 4th ed. McGraw-Hill, New York (1991).
- Reppich, M.; and S. Zagermann. "A New Design Method for Segementally Baffled Heat Exchangers." *Comput Chem Eng* **19 (Suppl)**: S137–S142 (1995).
- Sama, D. A. "Economic Optimum LMTD at Heat Exchangers." *AIChE National Meeting*. Houston, Texas (March 1983).
- Schweyer, H. E. *Process Engineering Economics*. McGraw-Hill, New York, (1955), p. 214.
- Steinmeyer, D. E. "Process Energy Conservation." *Kirk-Othmer Encyclopedia Supplemental Volume*, 3d ed. Wiley, New York (1984).
- Swearingen, J. S.; and J. E. Ferguson. "Optimized Power Recovery from Waste Heat." *Chem Eng Prog* August 66-70 (1983).
- Tarrer, A. R.; H. C. Lim; and L. B. Koppel. "Finding the Economically Optimum Heat Exchanger." *Chem Eng* **79**: 79–84 Oct. 4 (1971).
- Yanniotis, S.; and P. A. Pilavachi. "Mathematical Modeling and Experimental Validation of an Absorber-Driven Multiple Effect Evaporator." *Chem Eng Technol* **19**: 448–455 (1996).
- Zamora, J. M.; and I. E. Grossmann. "A Global MINLP Optimization Algorithm for the Synthesis of Heat Exchanger Networks with No Stream Splits." *Comput Chem Eng* **22**: 367–384 (1998).

SUPPLEMENTARY REFERENCES

- Chaudhuri, P. D.; U. M. Diweker; and J. S. Logsdon. "An Automated Approach for the Optimal Design of Heat Exchangers." *Ind Eng Chem Res* **36** (9): 3685–3693 (1999).
- Ciric, A. R.; and C. A. Floudas. "Heat Exchanger Network Synthesis Without Documentation." *Comput Chem Eng* **15**: 385–396 (1991).
- Colmenares, T. R.; and W. D. Seider. "Heat and Power Integration of Chemical Processes." *AIChE J* **33**: 898–915 (1987).
- Cornellisen, R. L.; and G. G. Hiss. "Thermodynamic Optimization of a Heat Exchanger." *Int J Heat Mass Transfer* **42** (5): 951–959 (1999).
- Daichendt, M. M.; and I. E. Grossmann. "Preliminary Screening Procedure for the MINLP Synthesis of Process Systems II. Heat Exchanger Networks." *Comput Chem Eng* **18**: 679–710 (1994).
- Duran, M. A.; and I. E. Grossmann. "Simultaneous Optimization and Heat Integration of Chemical Processes." *AIChE J* **32**: 123–138 (1986).
- Fabbri, G. "Heat Transfer Optimization in Internally Finned Tubes Under Laminar Flow Conditions." *Int J Heat Mass Transfer* **41** (10): 1243–1253 (1998).

- Georgiadis, M. C.; L. G. Papageorgiou; and S. Macchietto. "Optimal Cleaning Policies in Heat Exchanger Networks under Rapid Fouling." *Ind Eng Chem Res* **39** (2): 441–454 (2000).
- Gunderson, T.; and L. Naess. "The Synthesis of Cost Optimal Heat Exchanger Networks. An Industrial Review of the State of the Art." *Comput Chem Eng* **12**: 503–530 (1988).
- Ikegami, Y.; and A. Bejan. "On the Thermodynamic Optimization of Power Plants with Heat Transfer and Fluid Flow Irreversibilities." *J Solar Energy Engr* **120** (2): 139–144 (1998).
- Kalitventzeff, B. "Mixed Integer Nonlinear Programming and its Application to the Management of Utility Networks." *Eng Optim* **18**: 183–207 (1991).
- Lang, Y. D.; L. T. Biegler; and I. E. Grossmann. "Simultaneous Optimization and Heat Integration with Process Simulators." *Comput Chem Eng* **12**: 311–328 (1988).
- Linnhoff, B. "Pinch Analysis—A State-of-the-Art Overview." *Trans I Chem E* **71** (A): 503–523 (1993).
- Luus, R. "Optimization of Heat Exchanger Networks." *Ind Eng Chem Res* **32** (11): 2633–2635 (1993).
- Peterson, J.; and Y. Bayazitoglu. "Optimization of Cost Subject to Uncertainty Constraints in Experimental Fluid Flow and Heat Transfer." *J Heat Transfer* **113**: 314–320 (1991).
- Quesada, I.; and I. E. Grossmann. "Global Optimization Algorithm for Heat Exchanger Networks." *Ind Eng Chem Res* **32**: 487–499 (1993).
- Tayal, M. C.; Y. Fu; and U. M. Diwekar. "Optimal Design of Heat Exchangers: A Genetic Algorithm Framework." *Ind Eng Chem Res* **38** (2): 456–467 (1999).
- Yee, T. F.; I. E. Grossmann; and Z. Kravanja. "Simultaneous Optimization Models for Heat Integration—I. Area and Energy Targeting and Modeling of Multistream Exchangers." *Comput Chem Eng* **14**: 1165–1183 (1990).
- Yeh, R. "Errors in One-dimensional Fin Optimization Problem for Convective Heat Transfer." *Int J Heat Mass Transfer* **39** (14): 3075–3078 (1996).
- Zhu, X. X.; and N. D. K. Asante. "Diagnosis and Optimization Approach for Heat Exchanger Network Retrofit." *AIChE J* **45**: 1488–1503 (1999).

SEPARATION PROCESSES

Example

12.1 Optimal Design and Operation of a Conventional Staged-Distillation Column	443
12.2 Optimization of Flow Rates in a Liquid-Liquid Extraction Column	448
12.3 Fitting Vapor-Liquid Equilibrium Data Via Nonlinear Regression	451
12.4 Determination of the Optimal Reflux Ratio for a Staged-Distillation Column	453
References	458
Supplementary References	458

SEPARATIONS ARE AN important phase in almost all chemical engineering processes. Separations are needed because the chemical species from a single source stream must be sent to multiple destinations with specified concentrations. The sources usually are raw material inputs and reactor effluents; the destinations are reactor inputs and product and waste streams. To achieve a desired species allocation you must determine the best types and sequence of separators to be used, evaluate the physical or chemical property differences to be exploited at each separator, fix the phases at each separator, and prescribe operating conditions for the entire process. Optimization is involved both in the design of the equipment and in the determination of the optimal operating conditions for the equipment.

A wide variety of separation processes exist (Meloan, 1999), including

Centrifugation	Flotation
Chromatography	Freeze drying
Dialysis	Ion exchange
Distillation	Membranes
Electrophoresis	Osmosis
Extraction	Zone melting
Filtration	

Although each type of process is based on different physical principles, the mathematical models used to represent a process are surprisingly similar. Usually the equations are material or energy balances, either steady-state (most often) or dynamic, corresponding to fundamental laws, and empirical equilibrium relations. The equations may involve discrete or continuous variables depending on the simplifying assumptions made. For example, for a staged-distillation column the typical assumptions might include one or more of the following:

1. The hold-up liquid on each plate is completely mixed.
2. A constant hold-up exists on each plate, in the reboiler, and in the condenser-accumulator system.
3. The fluid dynamic response time is negligible.
4. The effects of pressure changes in various sections of the column on the physical properties of the system being distilled are negligible.
5. The saturated liquid and vapor enthalpies can be expressed as a linear function of compositions.
6. All fluid streams are single phase, and liquid entrainment and vapor hold up are negligible.
7. The column operates adiabatically; heat lost to the atmosphere is negligible.
8. The liquid and vapor compositions leaving a plate are a function only of the compositions in the column and experimental plate efficiencies, and can be described as a linear function of corrected compositions at various sections of the column.
9. At a constant operating steam pressure, the heat transfer in the reboiler is a function of composition.

Many of these assumptions are made to reduce the complexity of the mathematical model for the distillation process. Some may have negligible adverse effects in a specific process, whereas others could prove to be too restrictive.

This chapter contains examples of optimization techniques applied to the design and operation of two of the most common staged and continuous processes, namely, distillation and extraction. We also illustrate the use of parameter estimation for fitting a function to thermodynamic data.

EXAMPLE 12.1 OPTIMAL DESIGN AND OPERATION OF A CONVENTIONAL STAGED-DISTILLATION COLUMN

Distillation is probably the most widely used separation process in industry. Various classes of optimization problems for steady-state distillation are, in increasing order of complexity,

1. Determine the optimal operating conditions for an existing column to achieve specific performance at minimum cost (or minimum energy usage) given the feed(s). Usually, the manipulated (independent) variables are indirect heat inputs, cooling stream inputs, and product flow rates. The number of degrees of freedom is most likely equal to the number of product streams. Specific performance is measured by specified component concentrations or fractional recoveries from the feed (specifications leading to equality constraints) or minimum (or maximum) concentrations and recoveries (specifications leading to inequality constraints). In principle, any of the specified quantities as well as costs can be calculated from the values of the manipulated variables given the mathematical model (or computer code) for the column. When posed as described earlier, the optimization problem is a nonlinear programming problem often with implicit nested loops for calculation of physical properties. If the number of degrees of freedom is reduced to zero by specifications placed on the controlled variables, the optimization problem reduces to the classic problem of distillation design that requires just the solution of a set of nonlinear equations.
2. A more complex problem is to determine not only the values of the operating conditions as outlined in item 1 but also the (minimum) number of stages required for the separation. Because the stages are discrete (although in certain examples in this book we have treated them as continuous variables), the problem outlined in item 1 becomes a nonlinear mixed-integer programming problem (see Chapter 9). In this form of the design problem, the costs include both capital costs and operating costs. Capital costs increase with the number of stages and internal column flow rates, whereas operating costs decrease up to a certain point.
3. An even more difficult problem is to determine the number of stages and the optimal locations for the feed(s) and side stream(s) withdrawal. Fortunately, the range of candidates for stage locations for feed and withdrawals is usually small, and from a practical viewpoint the objective function is usually not particularly sensitive to a specific location within the appropriate range.

Optimization of distillation columns using mathematical programming, as opposed to other methods, has been carried out using many techniques, including search methods such as Hooke and Jeeves (Srygley and Holland, 1965), mixed-integer nonlinear programming (MINLP) (Frey et al., 1997; and Bauer and Stichlmair, 1998), genetic algorithms (Fraga and Matias, 1996), and successive quadratic programming (SQP) (Schmid and Biegler, 1994), which is the technique we use in this example. The review by Skogestad (1997) treats many of the various issues involved in the optimization of distillation columns beyond those we illustrate here.

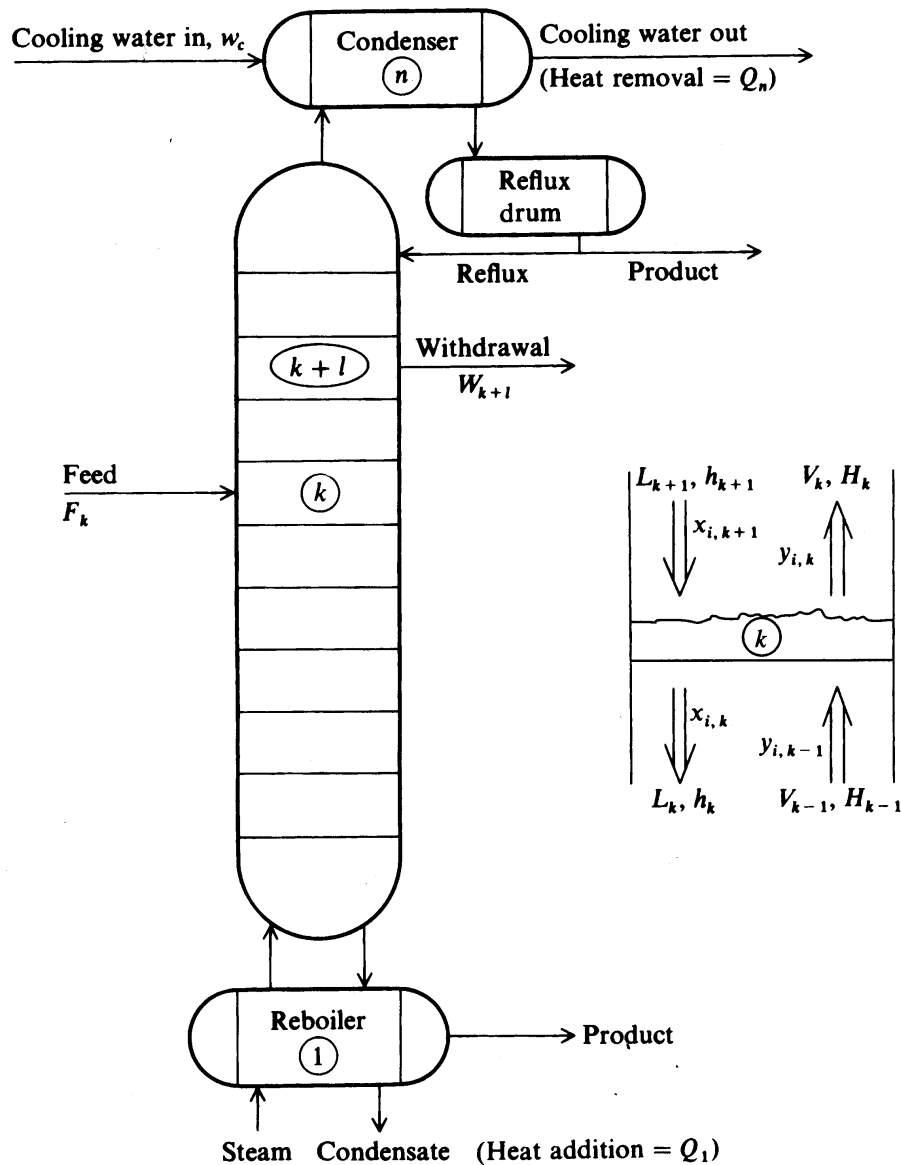


FIGURE E12.1
Schematic of a staged distillation column.

This example focuses on the design and optimization of a steady-state staged column. Figure E12.1 shows a typical column and some of the notation we will use, and Table E12.1A lists the other variables and parameters. Feed is denoted by superscript F . Withdrawals take the subscripts of the withdrawal stage. Superscripts V for vapor and L for liquid are used as needed to distinguish between phases. If we number the stages from the bottom of the column (the reboiler) upward with $k = 1$, then $V_0 = L_1 = 0$, and at the top of the column, or the condenser, $V_n = L_{n+1} = 0$. We first formulate the equality constraints, then the inequality constraints, and lastly the objective function.

The equality constraints. The process model comprises the equality constraints. For a conventional distillation column we have the following typical relations:

TABLE E12.1A
Notation for distillation example

F_k	flow of feed into stage k , moles
h_k	liquid enthalpy (a function of p_k , T_k , and \mathbf{x}_k) on stage k
H_k	vapor enthalpy (a function of p_k , T_k , and \mathbf{y}_k) on stage k
k	stage index number, $k = 1, \dots, n$.
$K_{i,k}$	equilibrium constant for component i for the mixture on stage k (a function of p_k , T_k , \mathbf{x}_k , \mathbf{y}_k)
L_k	flow of liquid from stage k , moles
m	number of components, $i = 1, \dots, m$
p_k	pressure on stage k
Q_k	heat transfer flow to stage k (positive when into stage)
T_k	temperature on stage k
V_k	flow of vapor from stage k , moles
W_k	withdrawal stream from stage k , moles
$x_{i,k}$	mole fraction of component i on stage k in the liquid phase
$y_{i,k}$	mole fraction of component i on stage k in the vapor phase

1. Total material balances (one for each stage k)

$$F_k^L + F_k^V + V_{k-1} + L_{k+1} = V_k + L_k + W_k^V + W_k^L \quad (a)$$

(F_k and W_k are ordinarily not involved in most of the stages)

2. Component material balances (one for each component i for each stage k)

$$x_{i,k}^F F_k^L + y_{i,k}^F F_k^V + y_{i,k-1} V_{k-1} + x_{i,k+1} L_{k+1} = y_{i,k} V_k + x_{i,k} L_k + y_{i,k} W_k^V + x_{i,k} W_k^L \quad (b)$$

3. Energy balance (one for each stage)

$$Q_k + h_k^F F_k + H_{k-1} V_{k-1} + h_{k+1} L_{k+1} = H_k V_k + h_k L_k + H_k W_k^V + h_k W_k^L \quad (c)$$

4. Equilibrium relations for liquid and vapor at each stage (one for each stage)

$$y_{i,k} = K_{i,k} x_{i,k} \quad (d)$$

5. Relation between equilibrium constant and p , T , x , y (one for each stage)

$$K_{i,k} = K_i(p_k, T_k, \mathbf{x}_k, \mathbf{y}_k) \quad (e)$$

6. Relation between enthalpies and p , T , x , y (one for each stage)

$$h_k = h(p_k, T_k, \mathbf{x}_k) \quad (f)$$

$$H_k = H(p_k, T_k, \mathbf{y}_k) \quad (g)$$

The preceding classic set of algebraic equations form a well-defined sparse structure that has been analyzed extensively. Innumerable techniques of solution have been proposed for problems with 0 degrees of freedom, that is, the column operating or design variables are completely specified.

Our interest here in posing an optimization problem is to have one or more degrees of freedom left after prespecifying the values of most of the independent variables. Frequently, values are given for the following parameters:

- (a) Number of stages
- (b) Flow rate, composition, and enthalpy of the feed(s)
- (c) Location of the feed(s) and side stream withdrawal(s)
- (d) Flow rate of the side stream(s)
- (e) Heat input rate to each stage except one
- (f) Stage pressures (based on column detailed design specifications)

Reactive distillation involves additional degrees of freedom (Mujtaba and Macchietto, 1997). If the controllable parameters remaining to be specified, namely (1) one heat input, and (2) the flow rate of the product (or the reflux ratio), are determined via optimization, all of the values of V_k , L_k , T_k , $x_{i,k}$, and $y_{i,k}$ and the enthalpies can be calculated. More than 2 degrees of freedom can be introduced by eliminating some of the prespecified parameters values.

7. Certain implicit equality constraints exist

Because of the way the model is specified, you must take into account the following additional equations as constraints in the column model:

$$\sum_{i=1}^m x_{i,k} = 1 \quad (h)$$

$$\sum_{i=1}^m y_{i,k} = 1 \quad (i)$$

The inequality constraints. Various kinds of inequality constraints exist, such as requiring that all of the $x_{i,k}$, $y_{i,k}$, Q_k , F_k , W_k , and so on be positive, that upper and lower bounds be imposed on some of the product stream concentrations, and specification of the minimum recovery factors. A recovery factor for stage k is the ratio

$$\frac{x_{i,k}W_k^L + y_{i,k}W_k^V}{\sum_i (x_{i,k}F_k^L + y_{i,k}F_k^V)}$$

The objective function. The main costs of operation are the heating and cooling costs that are related to Q_1 and Q_n , respectively. We assume all the other values of Q_k are zero. Q_n is determined from the energy balance, so that Q_1 is the independent variable. The cost of operation per annum is assumed to be directly proportional to Q_1 because the maintenance and cooling costs are relatively small and the capital costs per annum are already fixed. Consequently, the objective function is relatively simple:

$$\text{Minimize: } Q_1 \quad (j)$$

As posed here, the problem is a nonlinear programming one and involves nested loops of calculations, the outer loop of which is Equation (j) subject to Equations (a) through (i), and subject to the inequality constraints. If capital costs are to be included in the objective function, refer to Frey and colleagues (1997).

Results for a specific problem with 5 degrees of freedom. For illustration, we use the data of Sargent and Gaminibandara (1976) for the objective function (j).

The problem is to determine the location and individual amounts of the feeds given the following information.

A column of four stages exists analogous to that shown in Figure E12.1 except that more than one feed can exist (the reboiler is stage 1 and the condenser is stage 4). Feed and product specifications are

$$\text{Total feed} = 100 \text{ lb mol/h liquid}$$

$$h_F = 4000 \text{ Btu/lb mol}$$

$$x_1 = 0.05 \text{ (C}_3\text{H}_8\text{)}$$

$$x_2 = 0.15 \text{ (i-C}_4\text{H}_{10}\text{)}$$

$$x_3 = 0.25 \text{ (n-C}_4\text{H}_{10}\text{)}$$

$$x_4 = 0.20 \text{ (i-C}_5\text{H}_{12}\text{)}$$

$$x_5 = 0.35 \text{ (n-C}_5\text{H}_{12}\text{)}$$

$$\text{Top product} = 10 \text{ lb mol/h liquid}$$

$$x_5 \leq 0.07$$

The equality constraints are Equations (a)–(i) plus

$$\sum_{k=1}^4 F_k = 100 \quad (k)$$

The inequality constraints are ($k = 1, \dots, 4$)

$$Q_1 \geq 0 \quad (l)$$

$$Q_4 \leq 0 \quad (m)$$

$$x_{i,k} \geq 0 \quad (n)$$

$$y_{i,k} \geq 0 \quad (o)$$

$$F_k \geq 0 \quad (p)$$

$$x_{5,4} \leq 0.07 \quad (q)$$

This problem has 5 degrees of freedom, representing the five variables Q_1 , F_1 , F_2 , F_3 , and F_4 .

Various rules of thumb and empirical correlations exist to assist in making initial guesses for the values of the independent variables. All the values of the feeds here can be assumed to be equal initially. If the reflux ratio is selected as an independent variable, a value of 1 to 1.5 times the minimum reflux ratio is generally appropriate.

To solve the problem a sequential quadratic programming code was used in the outer loop of calculations. Inner loops were used to evaluate the physical properties. Forward-finite differences with a step size of $h = 10^{-7}$ were used as substitute for the derivatives. Equilibrium data were taken from Holland (1963). The results shown in Table E12.1B were essentially the same as those obtained by Sargent and Gaminibandara.

TABLE E12.1B
Results of optimization

Variable	Initial guess for the variable	Optimal values for the variable
F_1	25	23.7
F_2	25	0
F_3	25	0
F_4	25	76.3
Q_1	5.0×10^6	3.38×10^5
$x_{5,4}$	—	0.07

We can conclude that it is possible to use some of the cold feed as reflux in the top stage without voiding the product composition specification. This outcome is not an obvious choice for the problem specifications.

EXAMPLE 12.2 OPTIMIZATION OF FLOW RATES IN A LIQUID-LIQUID EXTRACTION COLUMN

Liquid-liquid extraction is carried out either (1) in a series of well-mixed vessels or stages (well-mixed tanks or in plate column), or (2) in a continuous process, such as a spray column, packed column, or rotating disk column. If the process model is to be represented with integer variables, as in a staged process, MILNP (Glanz and Stichlmair, 1997) or one of the methods described in Chapters 9 and 10 can be employed. This example focuses on optimization in which the model is composed of two first-order, steady-state differential equations (a plug flow model). A similar treatment can be applied to an axial dispersion model.

Figure E12.2a illustrates a typical steady-state continuous column. The model and the objective function are formulated as follows.

The process model. Under certain conditions, the plug flow model for an extraction process has an analytical solution. Under other conditions, numerical solutions of the equations must be used. As a practical matter, specifying the model so that an analytical solution exists means assuming that the concentrations are expressed on a solute-free mole basis, that the equilibrium relation between Y and X is a straight line $Y^* = mX + B$ (i.e., not necessarily through the origin), and that the operating line is straight, that is, the phases are insoluble. Then the model is

$$\frac{dX}{dZ} - N_{OX}(X - Y) = 0 \quad (a)$$

$$\frac{dY}{dZ} - FN_{OX}(X - Y) = 0 \quad (b)$$

where F = extraction factor (mv_X/v_Y)

m = distribution coefficient

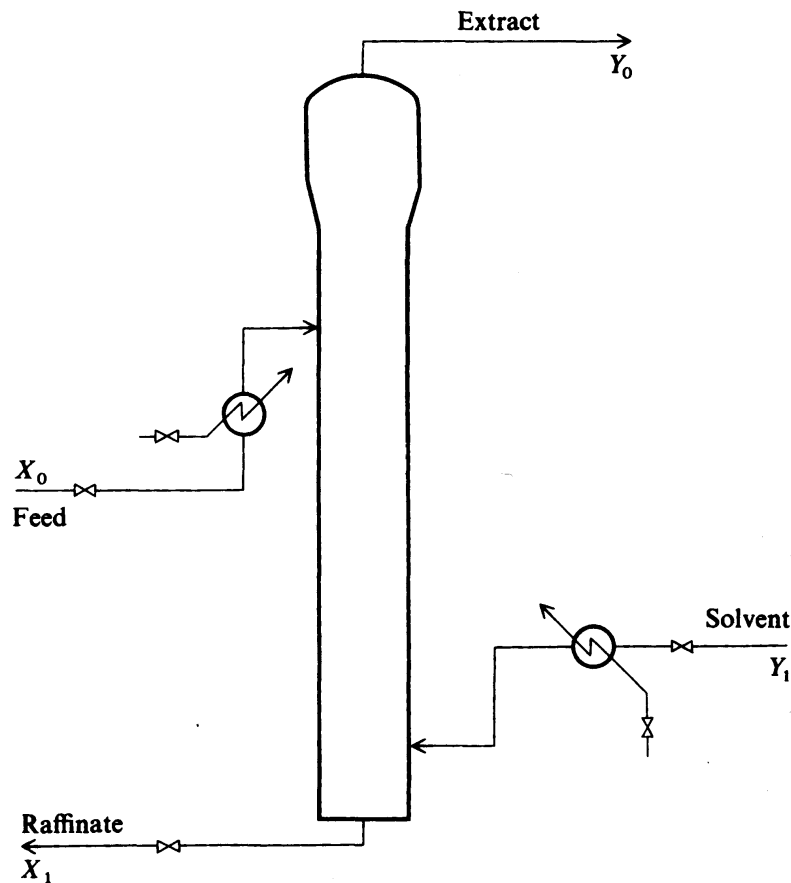
N_{OX} = number of transfer units

v_X, v_Y = superficial velocity in raffinate, extract phase, respectively

X = dimensionless raffinate phase concentration

Y = dimensionless extract phase concentration

Z = dimensionless contactor length

**FIGURE E12.2a**

Extraction column schematic for Example 12.2. (The internal rotating disks are not shown.)

Figure E12.2a shows the boundary conditions X_0 and Y_1 . Given values for m , N_{OX} , and the length of the column, a solution for Y_0 in terms of v_X and v_Y can be obtained; X_1 is related to Y_0 and F via a material balance: $X_1 = 1 - (Y_0/F)$. Hartland and Mecklenburgh (1975) list the solutions for the plug flow model (and also the axial dispersion model) for a linear equilibrium relationship, in terms of F :

$$Y_0 = \frac{F\{1 - \exp[N_{OX}(1 - F)]\}}{1 - F \exp[N_{OX}(1 - F)]} \quad (c)$$

In practice, N_{OX} is calculated from experimental data by least squares or from an explicit relation for the plug flow model.

$$N_{OX} = \left(\frac{1 - X_1}{X_1 + Y_0 - 1} \right) \ln \left(\frac{X_1}{1 - Y_0} \right) \quad (d)$$

Jackson and Agnew (1980) summarized a number of correlations for N_{OX} such as

$$N_{OX} = 4.81 \left(\frac{v_X}{v_Y} \right)^{0.24} \quad (e)$$

The value of $m = 1.5$.

Inequality constraints. Implicit constraints exist because of the use of dimensionless variables

$$X_0 \leq X \leq X_1$$

$$Y_1 \leq Y \leq Y_0 \quad (f)$$

Constraints on v_x and v_y are upper and lower bounds such as

$$0.05 < v_x < 0.25$$

$$0.05 < v_y < 0.30 \quad (g)$$

and the flooding constraint

$$v_x + v_y \leq 0.20 \quad (h)$$

Objective function. The objective function is to maximize the total extraction rate for constant disk rotation speed subject to the inequality and equality constraints:

$$\text{Maximize: } f = v_y Y_0 \quad (i)$$

Results of the optimization. Figure E12.2b illustrates contours of the objective function for the plug flow model; the objective function (i) was optimized by the GRG

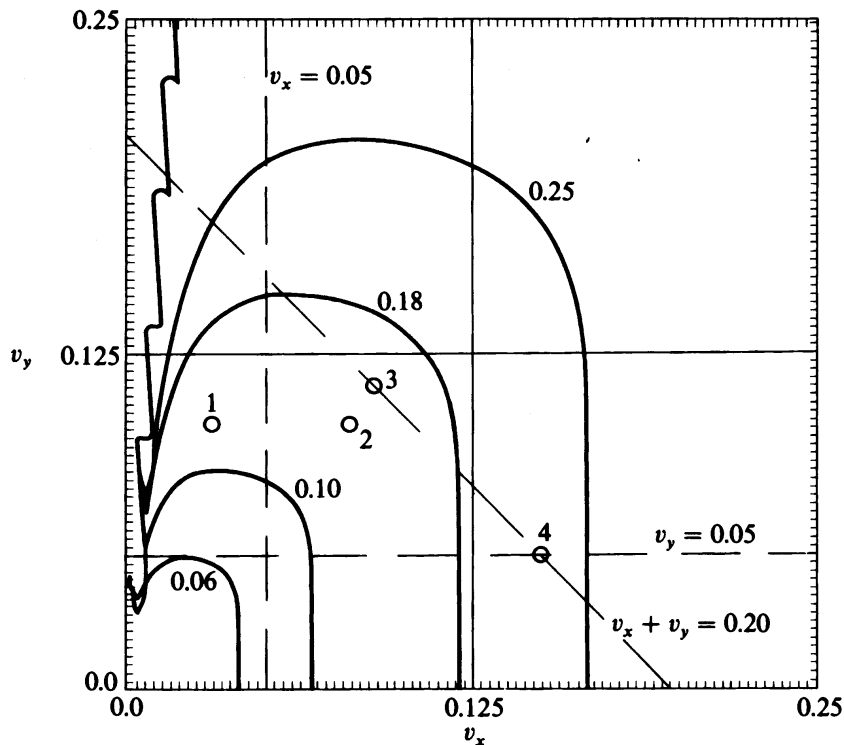


FIGURE E12.2b

Contours (the heavy lines) for the objective function of extraction process. Points 1, 2, 3, and 4 indicate the progress of the reduced-gradient method toward the optimum (point 4).

(generalized reduced-gradient) method. For small values of v_x (< 0.01), the contours drop off quite rapidly. The starting point (point 1)

$$v_x = 0.03$$

$$v_y = 0.10$$

is infeasible. Points 2, 3, and 4 in Figure E12.2b show the change as the vector of independent variables moves toward the optimum. Point 2 indicates the first feasible values of v_x and v_y (0.08, 0.10), point 3 indicates where the flooding constraint (h) is active, and point 4 is the constrained optimum (0.15, 0.05). The value of the objective function at point 4 is 0.225.

EXAMPLE 12.3 FITTING VAPOR–LIQUID EQUILIBRIUM DATA VIA NONLINEAR REGRESSION

Valid physical property relationships form an important feature of a process model. To validate a model, representative data must fit by some type of correlation using an optimization technique. Nonlinear regression instead of linear regression may be involved in the fitting. We illustrate the procedure in this example.

Separation systems include in their mathematical models various vapor–liquid equilibrium (VLE) correlations that are specific to the binary or multicomponent system of interest. Such correlations are usually obtained by fitting VLE data by least squares. The nature of the data can depend on the level of sophistication of the experimental work. In some cases it is only feasible to measure the total pressure of a system as a function of the liquid phase mole fraction (no vapor phase mole fraction data are available).

Vapor–liquid equilibria data are often correlated using two adjustable parameters per binary mixture. In many cases, multicomponent vapor–liquid equilibria can be predicted using only binary parameters. For low pressures, the equilibrium constraint is

$$x_i \gamma_i p_i^{\text{sat}} = y_i p \quad (i = 1, 2) \quad (a)$$

where p = the total pressure

p_i^{sat} = the saturation pressure of component i

x_i = the liquid phase mole fraction of component i

γ_i = the activity coefficient

y_i = the vapor phase mole fraction

The van Laar model for a binary mixture is

$$\ln \gamma_1 = A_{12} \left[\frac{A_{21} x_2}{A_{12} x_1 + A_{21} x_2} \right]^2 \quad (b)$$

and

$$\ln \gamma_2 = A_{21} \left[\frac{A_{12} x_1}{A_{12} x_1 + A_{21} x_2} \right]^2 \quad (c)$$

where A_{12} and A_{21} are binary constants that are adjusted by optimization to fit the calculated data for x_i . To use total pressure measurements we write

$$p = y_1 p + y_2 p \quad (d)$$

TABLE E12.3
Experimental VLE data for the system
(1) Water. (2) 1,4 dioxane at 20°C.

Experimental data		Predicted values		
x_1	$p^{\text{expt}}(\text{mmHg})$	p^{calc}	p	y^{calc}
0.00	28.10	28.10	0.00	0.0
0.10	34.40	34.20	-0.20	0.2508
0.20	36.70	36.95	0.25	0.3245
0.30	36.90	36.97	0.07	0.3493
0.40	36.80	36.75	-0.05	0.3576
0.50	36.70	36.64	-0.06	0.3625
0.60	36.50	36.56	0.06	0.3725
0.70	35.40	35.36	-0.04	0.3965
0.80	32.90	32.84	-0.06	0.4503
0.90	27.70	27.72	0.02	0.5781
1.00	17.50	17.50	0.00	1.0

Antoine constants: $\log p^{\text{sat}} = a_1 - \frac{a_2}{T + a_3}$		p^{sat} : mmHg		
		T : °C		
	a_1	a_2	a_3	Range
(1) Water	8.07131	1730.630	233.426	(1–100°C)
(2) 1,4 dioxane	7.43155	1554.679	240.337	(20–105°C)

Note: Data reported by Hororka et al. (1936).

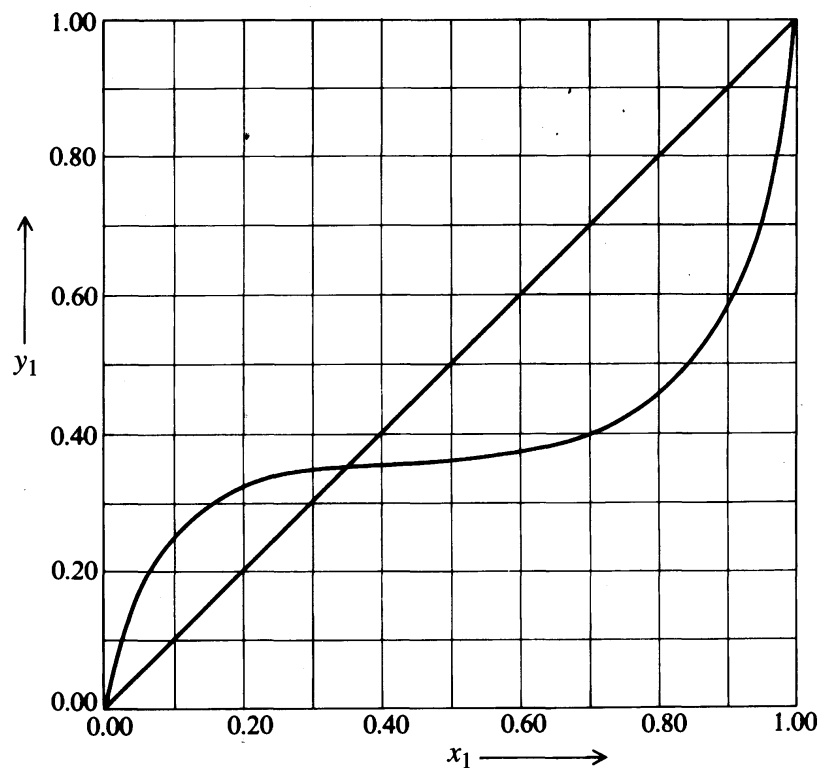


FIGURE E12.3

Experimental vapor–liquid equilibrium data, Example 12.3.

[Source: Gmehling et al. (1981).]

or, using Equations (a)–(c)

$$p = x_1 \exp \left[A_{12} \left(\frac{A_{21}x_2}{A_{12}x_1 + A_{21}x_2} \right)^2 \right] p_1^{\text{sat}} + x_2 \exp \left[A_{21} \left(\frac{A_{12}x_1}{A_{12}x_1 + A_{21}x_2} \right)^2 \right] p_2^{\text{sat}} \quad (e)$$

The saturation pressures can be predicted at a given temperature using the Antoine equation. For a given temperature and a binary system ($x_2 = 1 - x_1$)

$$p = p(x_1, A_{12}, A_{21}) \quad (f)$$

so that the two binary coefficients may be determined from experimental values of p versus x_1 by nonlinear least squares estimation (regression), that is, by minimizing the objective function

$$f = \sum_{i=1}^n (p_j^{\text{calc}} - p_j^{\text{expt}})^2 \quad (g)$$

where n is the number of data points.

In the book, *Vapor–Liquid Equilibrium Data Collection*, Gmehling and colleagues (1981), nonlinear regression has been applied to develop several different vapor–liquid equilibria relations suitable for correlating numerous data systems. As an example, p versus x_1 data for the system water (1) and 1,4 dioxane (2) at 20.00°C are listed in Table E12.3. The Antoine equation coefficients for each component are also shown in Table E12.3. A_{12} and A_{21} were calculated by Gmehling and colleagues using the Nelder–Mead simplex method (see Section 6.1.4) to be 2.0656 and 1.6993, respectively. The vapor phase mole fractions, total pressure, and the deviation between predicted and experimental values of the total p

$$\Delta p_j = p_j^{\text{calc}} - p_j^{\text{expt}}$$

are listed in Table E12.3 for increments of $x_1 = 0.10$. The mean Δp is 0.09 mmHg for pressures ranging from 17.5 to 28.10 mmHg. Figure E12.3 shows the predicted y_1 versus x_1 data; note that the model predicts an azeotrope at $x_1 = y_1 = 0.35$.

EXAMPLE 12.4 DETERMINATION OF THE OPTIMAL REFLUX RATIO FOR A STAGED-DISTILLATION COLUMN

Once a distillation column is in operation, the number of trays is fixed and very few degrees of freedom can be manipulated to minimize operating costs. The reflux ratio frequently is used to control the steady-state operating point. Figure E12.4a shows typical variable cost patterns as a function of the reflux ratio. The optimization of reflux ratio is particularly attractive for columns that operate with

1. High reflux ratio
2. High differential product values (between overhead and bottoms)
3. High utility costs
4. Low relative volatility
5. Feed light key far from 50 percent

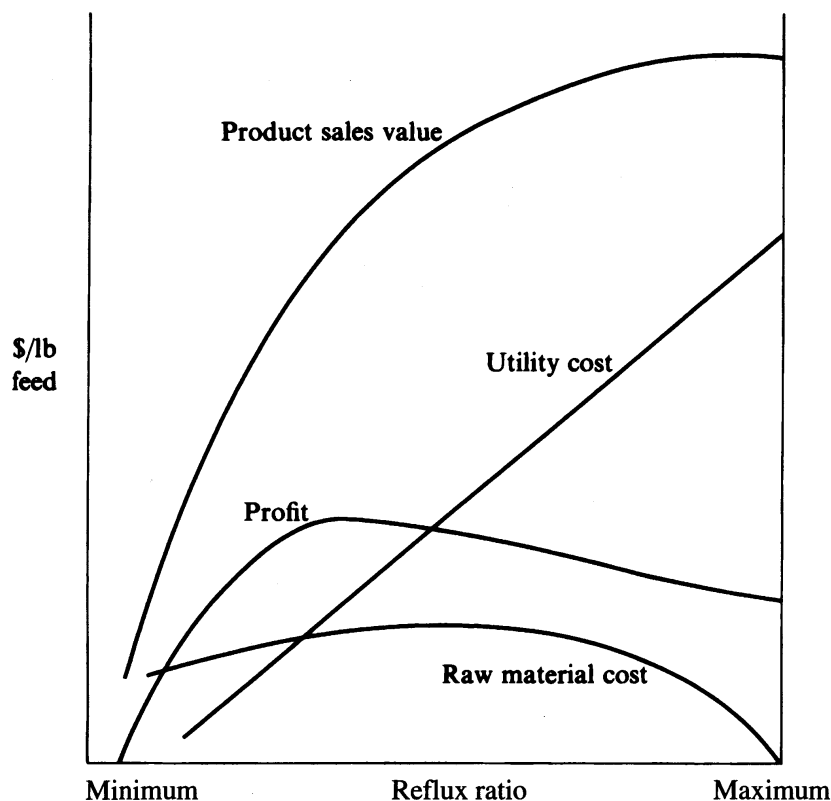


FIGURE E12.4a
Variable cost trade-offs for a distillation column.

In this example we illustrate the application of a one-dimensional search technique from Chapter 5 to a problem posed by Martin and coworkers (1981) of obtaining the optimal reflux ratio in a distillation column.

Martin and coworkers described an application of optimization to an existing tower separating propane and propylene. The lighter component (propylene) is more valuable than propane. For example, propylene and propane in the overhead product were both valued at \$0.20/lb (a small amount of propane was allowable in the overhead), but propane in the bottoms was worth \$0.12/lb and propylene \$0.09/lb. The overhead stream had to be at least 95 percent propylene. Based on the data in Table E12.4A, we will determine the optimum reflux ratio for this column using derivations provided by McAvoy (personal communication, 1985). He employed correlations for column performance (operating equations) developed by Eduljee (1975).

Equality constraints. The Eduljee correlation involves two parameters: R_m , the minimum reflux ratio, and N_m , the equivalent number of stages to accomplish the separation at total reflux. His operating equations relate N , α , X_F , X_D , and X_B (see Table E12.4A for notation) all of which have known values except X_B as listed in Table E12.4A. Once R is specified, you can find X_B by sequential solution of the three following equations.

TABLE E12.4A
Notation and values for the propane–propylene splitter

Symbol	Description	Value
B	Bottoms flow rate	
C_1	Reboiler heat cost	\$3.00/10 ⁶ Btu
C_2	Condenser cooling cost	\$0.00/10 ⁶ Btu
C_B	Value of propylene in bottoms	
C'_B	Value of propane in bottoms	
C_F	Cost per pound of propylene	
C'_F	Cost per pound of propane	
C_D	Value of propylene in overhead	
C'_D	Value of propane in overhead	
D	Distillate flow rate	
F	Feed rate	1,200,000 lb/day
L	Liquid flow rate	function of R (mol/day)
N	Number of equilibrium stages	94
N_m	Minimum equilibrium stages	function of reflux ratio, R
Q_C	Condenser load requirement	$Q_C \approx \lambda V$
Q_R	Reboiler heat requirement	$Q_R \approx \lambda V$
R	Reflux ratio	(To be optimized)
R_m	Minimum reflux ratio	11.17
U	Heavy key differential value	−\$0.08/lb
V	Vapor flow rate	function of R (mol/day)
W	Light key differential value	\$0.11/lb
X_B	Bottom light key mole fraction	(To be optimized)
X_D	Overhead light key mole fraction	0.95
X_F	Feed light key mole fraction	0.70
α	Relative volatility	1.105
λ	Latent heat	130 Btu/lb (avg. mixture)

First, calculate R_m

$$R_m = \frac{1}{(\alpha - 1)} \left[\frac{X_D}{X_F} - \alpha \frac{(1 - X_D)}{(1 - X_F)} \right] \quad (a)$$

Substitute the value of R_m in Equation (b) to find N_m

$$\left(\frac{N - N_m}{N + 1} \right) = 0.75 \left[1 - \left(\frac{R - R_m}{R + 1} \right)^{0.5668} \right] \quad (b)$$

Lastly, compute X_B from

$$N_m = \frac{\ln \{ [X_D / (1 - X_D)] \cdot [(1 - X_B) / X_B] \}}{\ln \alpha} \quad (c)$$

Equations (a)–(c) comprise equality constraints relating X_B and R .

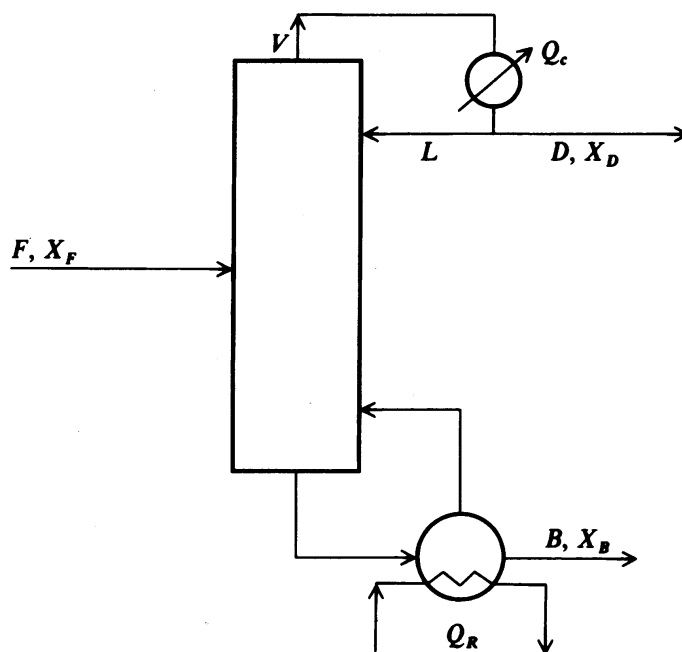


FIGURE E12.4b
Distillation column flow chart.

Once X_B is calculated, the overall material balance for the column shown in Figure E12.4b can be computed. The pertinent equations are (the units are moles)

$$F = D + B \quad (d)$$

$$X_F F = X_D D + X_B B \quad (e)$$

Equations (d) and (e) contain two unknowns: D and B , which can be determined once F , X_F , X_B , and X_D are specified. In addition, if the assumption of constant molal overflow is made, then the liquid L and vapor flows V are

$$L = RD \quad (f)$$

$$V = (R + 1)D \quad (g)$$

Objective function. Next we develop expressions for the income and operating costs. The operating profit f is given by

$$f = \text{Propylene sales} + \text{Propane sales} - \text{Utility costs} - \text{Raw material costs} \quad (h)$$

$$f = (C_D X_D D + C_B X_B B) + [C'_D(1 - X_D)D + C'_B(1 - X_B)B] \\ - [C_1 Q_R + C_2 Q_C] - [C_F X_F F + C'_F(1 - X_F)F] \quad (i)$$

The brackets [] indicate the correspondence between the words in Equation (h) and the symbols in Equation (i). Q_R is the reboiler heat requirement and Q_C is the cooling load.

Equation (i) can be rearranged by substituting for DX_D in the propylene sales and for BX_B in the propane sales using Equation (e) and defining $-W = C_B - C_D$ and $-U = C'_D - C'_B$ as follows

TABLE E12.4B
Iterations in quadratic interpolation test problem

Iteration	Left bracket		Center point		Right bracket		Interpolated values	
	x	f	x	f	x	f	x	f
1	16.00	3967.13	18.00	3922.14	20.00	4256.45	17.24	3872.22
2	18.00	3922.14	17.24	3872.22	16.00	3967.13	17.16	3870.79
3	17.24	3872.22	17.16	3870.79	16.00	3967.13	17.09	3870.21
4	17.16	3870.79	17.09	3870.21	16.00	3967.13	17.06	3870.18
5	17.09	3870.21	17.06	3870.18	16.00	3967.13	17.06	3870.17

Final solution
 $x = 17.06$
 $f = 3870.17$

TABLE E12.4C
Sensitivity study at the reflux ratio optimum

Reflux ratio (R)	X_B (mol fraction)	Costs (\$/day)
17.07	0.0432	3870
18.77*	0.0303	4024
15.36*	0.0683	4159

*Indicates $17.07 \pm 10\%$

$$f = C_D X_F F + C_B'(1 - X_F)F - C_F X_F F - C_F'(1 - X_F)F - C_1 Q_R - C_2 Q_C - W X_B B - U(1 - X_D)D \quad (j)$$

Note that the first four terms of f are fixed values, hence these terms can be deleted from the expression for f in the optimization. In addition, it is reasonable to assume $Q_R \approx Q_C \approx \lambda V$. Lastly, the right-hand side of Equation (j) can be multiplied by -1 to give the final form of the objective function (to be minimized):

$$f_1 = (C_1 + C_2)\lambda V + W X_B B + U(1 - X_D)D \quad (k)$$

Note: λ must be converted to Btu/mol, and the costs to \$/mol.

Solution. Based on the data in Table E12.4A we minimized f_1 with respect to R using a quadratic interpolation one-dimensional search (see Chapter 5). The value of R_m from Equation (a) was 11.338. The initial bracket was $12 \leq R \leq 20$, and $R = 16, 18$, and 20 were selected for the initial three points. The convergence tolerance on the optimum required that f_1 should not change by more than 0.01 from one iteration to the next.

The iterative program incorporating the quadratic interpolation search yielded the results in Table E12.4B. The optimum reflux ratio was 17.06 and the cost, f_1 , was \$3870/day. Table E12.4C shows the variation in f_1 for ± 10 percent change in R . The profit function changes \$100/day or more.

REFERENCES

- Bauer, M. H.; and J. Stichlmair. "Design and Economic Optimization of Azeotropic Distillation Process Using Mixed-Integer Nonlinear Programming." *Comput Chem Eng* **22**: 1271–1286 (1998).
- Eduljee, H. E. "Equations Replace Gilliland Plot." *Hydrocarbon Process* September: 120–124 (1975).
- Fraga, E. S.; and T. R. S. Matias. "Synthesis and Optimization of a Nonideal Distillation System Using a Parallel Genetic Algorithm." *Comput Chem Eng* **20 (Suppl)**: S79–S84 (1996).
- Frey, Th.; M. H. Bauer; and J. Stichlmair. "MINLP Optimization of Complex Columns for Azeotropic Mixtures." *Comput Chem Eng* **21 (Suppl)**: S217–S222 (1997).
- Glanz, S.; and J. Stichlmair. "Mixed Integer Optimization of Combined Solvent Extraction and Distillation Processes." *Comput Chem Eng* **21 (Suppl)**: S547–S552 (1997).
- Gmehling, T.; U. Onken; and W. Arlt. "Vapor–Liquid Equilibrium Data Collection." *DECHEMA VI*, Part 1A (1981).
- Hartland, S.; and J. C. Mecklenburgh. *The Theory of Backmixing*. Wiley, New York, Chapter 10 (1975).
- Holland, C. D. *Multicomponent Distillation*. Prentice-Hall, Englewood Cliffs, NJ (1963) p. 494.
- Hororka, F.; R. A. Schaefer; and D. Dreisbach. *J Am Chem Soc* **58**: 2264 (1936).
- Jackson, P. J.; and J. B. Agnew. "A Model Based Scheme for the On-Line Optimization of a Liquid Extraction Process." *Comput Chem Eng* **4**: 241 (1980).
- Martin, G. D.; P. R. Latour; and L. A. Richard. "Closed-Loop Optimization of Distillation Energy." *Chem Eng Prog* Sept: 33 (1981).
- Meloan, C. E. *Chemical Separations*. Wiley, New York (1999).
- Mujtaba, I. M.; and S. Macchietto. "Efficient Optimization of Batch Distillation with Chemical Reaction Using Polynomial Curve Fitting Techniques." *Ind Eng Chem Res* **36**: 2287–2295 (1997).
- Sargent, R. W. H.; and K. Gaminibandara. "Optimum Design of Plate Distillation Columns." In *Optimization in Action*, L. D. W. Dixon, ed. Academic Press, New York (1976).
- Schmid, C.; and L. T. Biegler. "Reduced Hessian Successive Quadratic Programming for Real Time Optimization." *Proceed IFAC Adv Control Chem Processes*, Kyoto, Japan, 173–178 (1994).
- Skogestad, S. "Dynamic and Control of Distillation Columns—A Critical Survey." *Modeling Identification Control* **18**: 177–217 (1997).
- Srygley, J. M.; and C. D. Holland. "Optimal Design of Conventional and Complex Columns." *AIChEJ* **11**: 695–701 (1965).

SUPPLEMENTARY REFERENCES

- Duennebier, G.; and C. C. Pantelides. "Optimal Design of Thermally Coupled Distillation Columns." *Ind Eng Chem Res* **38** (1): 162–176 (1999).
- El-Halwagi, M.; and V. Manousiouthakis. "Synthesis of Mass Exchange Networks." *AIChE J* **35**: 1233–1244 (1989).
- Floudas, C. A. "Separation Synthesis of Multicomponent Feed Streams into Multicomponent Product Streams." *AIChE J* **33**: 540–550 (1987).

- Floudas, C. A.; and G. E. Paules. "A Mixed-Integer Nonlinear Programming Formulation for the Synthesis of Heat-Integrated Distillation Sequences." *Comput Chem Eng* **12**: 531–546 (1988).
- Harding, S. T.; and C. A. Floudas. "Locating all Heterogeneous and Reactive Azeotropes in Multicomponent Mixtures." *Ind Eng Chem Res* **39** (6): 1576–1595 (2000).
- Logsdon, J. S.; and L. T. Biegler. "Accurate Determination of Optimal Reflux Policies for the Maximum Distillate Problem in Batch Distillation." *Ind Eng Chem Res* **32** (4): 692–700 (1993).
- Natarajan, V.; B. W. Bequette; and S. M. Cramer. "Optimization of Ion-Exchange Displacement Separations. I. Validation of an Iterative Scheme and its Use as a Methods Development Tool." *J Chromatogr A*. **876** (1 + 2): 51–62 (2000).
- Viswanathan, J.; and I. E. Grossmann. "An Alternate MINLP Model for Finding the Number of Trays Required for a Specified Separation Objective." *Comput Chem Eng* **17**: 949–955 (1993).
- Viswanathan, J.; and I. E. Grossmann. "Optimal Feed Locations and Number of Trays for Distillation Columns with Multiple Feeds." *Ind Eng Chem Res* **32**: 2942–2949 (1993).
- Wajge, R. M.; and G. V. Reklaitis. "An Optimal Campaign Structure for Multicomponent Batch Distillation with Reversible Reaction." *Ind Eng Chem Res* **37** (5): 1910–1916 (1998).
- Westerberg, A. W.; and O. Wahnschafft. "Synthesis of Distillation-Based Separation Processes." In *Advances in Chemical Engineering*, Vol. 23, *Process Synthesis*, J. L. Anderson, ed. Academic Press, New York (1996), pp. 63–170.

FLUID FLOW SYSTEMS

Example

13.1 Optimal Pipe Diameter	461
13.2 Minimum Work of Compression	464
13.3 Economic Operation of a Fixed-Bed Filter	466
13.4 Optimal Design of a Gas Transmission Network	469
References	478
Supplementary References	478

OPTIMIZATION OF FLUID flow systems encompasses a wide-ranging scope of problems. In water resources planning the objective is to decide what systems to improve or build over a long time frame. In water distribution networks and sewage systems, the time frame may be quite long, but the water and sewage flows have to balance at the network nodes. In pipeline design for bulk carriers such as oil, gas, and petroleum products, specifications on flow rates and pressures (including storage) must be met by suitable operating strategies in the face of unusual demands. Simpler optimization problems exist in which the process models represent flow through a single pipe, flow in parallel pipes, compressors, heat exchangers, and so on. Other flow optimization problems occur in chemical reactors, for which various types of process models have been proposed for the flow behavior, including well-mixed tanks, tanks with dead space and bypassing, plug flow vessels, dispersion models, and so on. This subject is treated in Chapter 14.

Optimization (and modeling) of fluid flow systems can be put into three general classes of problems: (1) the modeling and optimization under steady-state conditions, (2) the modeling and optimization under dynamic (unsteady-state) conditions, and (3) stochastic modeling and optimization. All three classes of problems are complicated for large systems. Under steady-state conditions, the principal difficulties in obtaining the optimum for a large system are the complexity of the topological structure, the nonlinearity of the objective function, the presence of a large number of possibly nonlinear inequality constraints, and the large number of variables. We do not consider optimization of dynamic or stochastic processes in this chapter. Instead, we focus on relatively simple steady-state fluid flow processes using the following examples:

1. Optimal pipe diameter for an incompressible fluid (Example 13.1)
2. Minimum work of gas compression (Example 13.2)
3. Economic operation of a fixed-bed filter (Example 13.3)
4. Optimal design of a gas transmission line (Example 13.4)

EXAMPLE 13.1 OPTIMAL PIPE DIAMETER

Example 2.8 briefly discussed how to determine the optimal flow in a pipe. In this example we consider how the trade-off between the energy costs for transport and the investment charges for flow in a pipe determines the optimum diameter of a pipeline. With a few simplifying assumptions, you can derive an analytical formula for the optimal pipe diameter and the optimal velocity for an incompressible fluid with density ρ and viscosity μ . In developing this formula the investment charges for the pump itself are ignored because they are small compared with the pump operating costs, although these could be readily incorporated in the analysis if desired. The mass flow rate m of the fluid and the distance L the pipeline is to traverse are presumed known, as are ρ and μ . The variables whose values are unknown are D (pipe diameter), Δp (fluid pressure drop), and v (fluid velocity); the optimal values of the three variables are to be determined so as to minimize total annual costs. Not all of the variables are independent, as you will see.

Total annual costs comprise the sum of the pipe investment charges and the operating costs for running the pump. Let C_{inv} be the annualized charges for the pipe and C_{op} be the pump operating costs. We propose that

$$C_{\text{inv}} = C_1 D^n L \quad (a)$$

$$C_{\text{op}} = \frac{C_0 m \Delta p}{\rho \eta} \quad (b)$$

where n = an exponent from a cost correlation (assumed to be 1.3)
 η = the pump efficiency
 C_0 and C_1 = cost coefficients

C_1 includes the capitalization charge for the pipe per unit length, and C_0 corresponds to the power cost (\$/kWh) due to the pressure drop. The objective function becomes

$$C = C_{\text{inv}} + C_{\text{op}} = C_1 D^n L + \frac{C_0 m \Delta p}{\rho \eta} \quad (c)$$

Note that Equation (c) has two variables: D and Δp . However, they are related through a fluid flow correlation as follows (part of the process model):

$$\Delta p = \frac{2f\rho v^2 L}{D} \quad (d)$$

where f is the friction factor. Two additional unspecified variables exist in Equation (d), namely v and f . Both m and f are related to v as follows:

$$m = \left(\frac{\rho \pi D^2}{4} \right) v \quad (e)$$

$$f = 0.046 \text{Re}^{-0.2} = \frac{0.046 \mu^{0.2}}{D^{0.2} v^{0.2} \rho^{0.2}} \quad (f)$$

Equation (e) is merely a definition of the mass flow rate. Equation (f) is a standard correlation for the friction factor for turbulent flow. (Note that the correlation between f and the Reynold's number (Re) is also available as a graph, but use of data from a graph requires trial-and-error calculations and rules out an analytical solution.)

To this point we isolated four variables: D , v , Δp , and f , and have introduced three equality constraints—Equations (d), (e), and (f)—leaving 1 degree of freedom (one independent variable). To facilitate the solution of the optimization problem, we eliminate three of the four unknown variables (Δp , v , and f) from the objective function using the three equality constraints, leaving D as the single independent variable. Direct substitution yields the cost equation

$$C = C_1 D^{1.3} L + 0.142 \frac{C_0}{\eta} m^{2.8} \mu^{0.2} \rho^{-2.0} D^{-4.8} L \quad (g)$$

Here, C_0 is selected with units $\{(\$/\text{year})/[(\text{lb}_m)(\text{ft}^2/\text{s}^3)]\}$. We can now differentiate C with respect to D and set the resulting derivative to zero

$$\frac{dC}{dD} = 0 = 1.3C_1D^{0.3}L - 0.682\frac{C_0}{\eta g_c}m^{2.8}\mu^{0.2}\rho^{-2.0}D^{-5.8} \quad (h)$$

and solve for D^{opt} :

$$D^{\text{opt}} = 0.900\left(\frac{C_0}{C_1\eta g_c}\right)^{0.164}m^{0.459}\mu^{0.033}\rho^{-0.328} \quad (i)$$

Note that L does not appear in the result.

Equation (i) permits a quick analysis of the optimum diameter as a function of a variety of physical properties. From the exponents in Equation (i), the density and mass flow rate seem to be fairly important in determining D^{opt} , but the ratio of the cost factors is less important. A doubling of m changes the optimum diameter by a factor of 1.4, but a doubling of the density decreases D^{opt} by a factor of 1.25. The viscosity is also not too important. For very viscous fluids, larger diameters resulting in lower velocities are indicated, whereas gases (low density) give smaller diameters and higher velocities. The validity of Equation (i) for gases is questionable, because the variation of gas velocity with pressure must be taken into account.

Using Equation (e)

$$v = \frac{4m}{\pi\rho D^2} \quad (j)$$

we can discover how the optimum velocity varies as a function of m , ρ , and μ by substituting Equation (i) for D^{opt} into (j):

$$v^{\text{opt}} = C_2m^{0.082}\mu^{-0.066}\rho^{-0.344} \quad (k)$$

where C_2 is a consolidated constant. Consider the effect of ρ on the optimum velocity. Generally optimum velocities for liquids vary from 3 to 8 ft/s, whereas for gases the range is from 30 to 60 ft/s. Although D^{opt} is influenced noticeably by changes in m , v^{opt} is very insensitive to changes in m .

Suppose a flow problem with the following specifications is posed:

$$\begin{aligned} m &= 50 \text{ lb/s} \\ \rho &= 60 \text{ lb/ft}^3 \\ \mu &= 6.72 \times 10^{-4} \text{ lb/(ft)(s)} \\ \eta &= 0.6 \text{ (60\% pump efficiency)} \end{aligned}$$

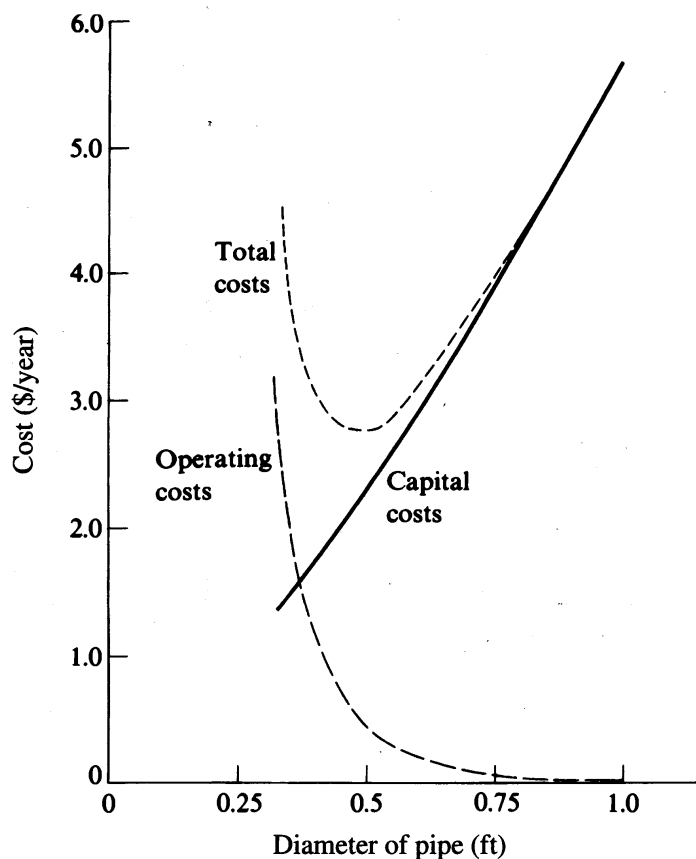
Purchased cost of electricity = \$0.05/kWh

8760 h/year of operation (100% stream factor)

$C_1 = \$5.7$ (D in ft); C_1D^n is an annualized cost expressed as $\$/(\text{ft})(\text{year})$

$L = \text{immaterial}$

The units in Equation (g) must be made consistent so that C is in dollars per year. For \$0.05/kWh, $C_0 = \$0.5938 \{(\$/\text{year})/[(\text{lb}_m)(\text{ft}^2/\text{s}^3)]\}$. Substitution of the values specified into Equation (i) gives $D^{\text{opt}} = 0.473 \text{ ft} = 5.7 \text{ in}$. The standard pipe schedule

**FIGURE E13.1**

Investment, operating, and total costs for pipeline example ($L = 1$ ft).

40 size closest to D^{opt} is 6 in. For this pipe size (ID = 6.065 in.) the optimum velocity is 4.2 ft/s. (A schedule 80 pipe has an ID of 5.7561 in.) Figure E13.1 shows the respective contributions of operating and investment costs to the total value of C .

As the process model is made more accurate and complicated, you can lose the possibility of obtaining an analytical solution of the optimization problem. For example, if (1) the pressure losses through the pipe fittings and valves are included in the model, (2) the pump investment costs are included as a separate term with a cost exponent (\bar{n}) that is not equal to 1.0, (3) elevation changes must be taken into account, (4) contained solids are present in the flow, or (5) significant changes in density occur, the optimum diameter will have to be calculated numerically.

EXAMPLE 13.2 MINIMUM WORK OF COMPRESSION

In this example we describe the calculation of the minimum work for ideal compressible adiabatic flow using two different optimization techniques, (a) analytical, and (b) numerical. Most real flows lie somewhere between adiabatic and isothermal flow. For adiabatic flow, the case examined here, you cannot establish a priori the relationship between pressure and density of the gas because the temperature is unknown as a function of pressure or density, hence the relation between pressure and

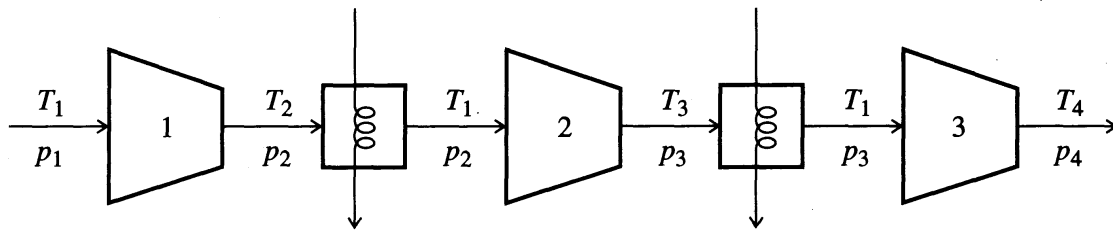


FIGURE E13.2

density is derived using the mechanical energy balance. If the gas is assumed to be ideal, and $k = C_p/C_v$ is assumed to be constant in the range of interest from p_1 to p_2 , you can make use of the well-known relation

$$pV^k = \text{Constant} \quad (a)$$

in getting the theoretical work per mole (or mass) of gas compressed for a single-stage compressor (McCabe and colleagues, 1993)

$$W = \frac{kRT_1}{k-1} \left[\left(\frac{p_2}{p_1} \right)^{(k-1)/k} - 1 \right] \quad (b)$$

where T_1 is the inlet gas temperature and R the ideal gas constant ($p_1 \hat{V}_1 = RT_1$). For a three-stage compressor with intercooling back to T_1 between stages as shown in Figure E13.2, the work of compression from p_1 to p_4 is

$$\hat{W} = \frac{kRT_1}{k-1} \left[\left(\frac{p_2}{p_1} \right)^{(k-1)/k} + \left(\frac{p_3}{p_2} \right)^{(k-1)/k} + \left(\frac{p_4}{p_3} \right)^{(k-1)/k} - 3 \right] \quad (c)$$

We want to determine the optimal interstage pressures p_2 and p_3 to minimize \hat{W} keeping p_1 and p_4 fixed.

Analytical solution. We set up the necessary conditions using calculus and also test to ensure that the extremum found is indeed a minimum.

$$\frac{\partial \hat{W}}{\partial p_2} = 0 = RT_1 \left[(p_1)^{(1-k)/k} (p_2)^{1/k} - (p_3)^{(k-1)/k} (p_2)^{(1-2k)/k} \right] \quad (d)$$

$$\frac{\partial \hat{W}}{\partial p_3} = 0 = RT_1 \left[(p_2)^{(1-k)/k} (p_3)^{1/k} - (p_4)^{(k-1)/k} (p_3)^{(1-2k)/k} \right] \quad (e)$$

The simultaneous solution of Equations (d) and (e) yields the desired results

$$p_2^2 = p_1 p_3 \quad \text{and} \quad p_3^2 = p_2 p_4$$

so that the optimal values of p_2 and p_3 in terms of p_1 and p_4 are

$$p_2^* = (p_1^2 p_4)^{1/3} \quad (f)$$

$$p_3^* = (p_4^2 p_1)^{1/3} \quad (g)$$

With these conditions for pressure, the work for each stage is the same.

To check the sufficiency conditions, we examine the Hessian matrix of \hat{W} (after substituting p_2^* and p_3^*) to see if it is positive-definite.

$$\nabla^2 W = RT_1 \left(\frac{k-1}{k} \right) \cdot \begin{bmatrix} 2[(p_1^*)^{(1-5k)/3k}][(p_4^*)^{-(1+k)/3k}] & [(p_1^*)^{(1-4k)/3k}][(p_4^*)^{-(1+2k)/3k}] \\ [(p_1^*)^{(1-4)/3k}][(p_2^*)^{-(1+2k)/3k}] & 2[(p_1^*)^{(1-3k)/3k}][(p_4^*)^{-(1+2k)/3k}] \end{bmatrix}$$

The two principal minors (the two diagonal elements) must be positive because p_1^* and p_4^* are both positive, and the determinant of $\nabla^2 \hat{W}$

$$4 \left[\frac{RT_1(k-1)}{k} \right]^2 [(p_1^*)^{(2-8k)/3k} (p_4^*)^{-(2+4k)/3k}] - \left[\frac{RT_1(k-1)}{k} \right]^2 [(p_1^*)^{2-8k/3k} (p_4^*)^{-(2+4k)/3k}] > 0$$

is also positive, hence $\nabla^2 \hat{W}$ is positive-definite.

Numerical solution. Numerical methods of solution do not produce the general solution given by Equations (f) and (g) but require that specific numerical values be provided for the parameters and give specific results. Suppose that $p_1 = 100$ kPa and $p_4 = 1000$ kPa. Let the gas be air so that $k = 1.4$. Then $(k-1)/k = 0.286$. Application of the BFGS algorithm to minimize \hat{W} in Equation (c) as a function of p_2 and p_3 starting with $p_2 = p_3 = 500$ yields

$$p_2^* = 215.44 \quad \text{compared with} \quad p_2^* = 215.44 \text{ from Equation (f)}$$

$$p_3^* = 464.17 \quad \text{compared with} \quad p_3^* = 464.16 \text{ from Equation (g)}$$

EXAMPLE 13.3 ECONOMIC OPERATION OF A FIXED-BED FILTER

Various rules of thumb exist for standard water filtration rates and cycle time before backwashing. Higher filtration rates may appear to be economically justified, however, when the filter loading is within conventional limits. In this example, we examine the issues involved for constant-rate filtration for a dual-media bed. Dual- and mixed-media beds result in increased production of water in a filter for two reasons. First, the larger grains (say charcoal approximately 1-mm size) as a top layer help reduce cake formation and deposition within the small (150-mm) top layer of the bed. Second, the head loss in the region of significant filtration is reduced.

With respect to the objective function for a filter, the total annual cost of filtration f is assumed to be the sum of the annualized capital costs f_c and the annual operating costs f_0 . The annualized capital cost is related to the cross-sectional area of the filter by the relation

$$f_c = rbA^z \tag{a}$$

where r = the capital recovery factor involving the discount rate and economic life of the filter

b = an empirical constant

- z = an empirical exponent
 A = the cross-sectional area of the filter

The cross-sectional area can be calculated by dividing the design flow rate by a quantity that is equal to the number of filter runs per day times the net water production per run per cross-sectional area:

$$A = \frac{q}{1440/[(V_f/Q) + t_b] \cdot (V_f - V_b)} \quad (b)$$

- where q = the design flow rate in gal/day, L/day (dual units given here)
 V_f = the volume of water filtered per unit area of bed per filter run in gal/ft², L/m²
 V_b = the volume of filtered water used for backwash per unit area of bed in gal/ft²; L/m²
 Q = the filtration rate in gal/(min)(ft²); L/(min)(m²)
 t_b = the filter down time for backwash, min
 1440 = the number of minutes/day

For a constant filtration rate, the length of the filter run is given by $t_f = V_f/Q$.

The water production per filter run V_f is based on a relation proposed by Letterman (1980) that assumes minimal surface cake formation by the time filtration is stopped because of head loss:

$$V_f = \frac{K_\rho D}{\beta C_0 n} \sum_{i=1}^n \log \frac{n \Delta H}{k_i D Q} \quad (c)$$

- where K_ρ = a constant related to the density of the deposit within the bed
 D = the overall depth of the bed, ft.
 β = the overall fraction of the influent suspended solids removed during the entire filter run
 C_0 = suspended solids concentration in the filter influent
 n = the number of layers $i = 1, \dots, n$ into which the filter is divided for use of Equation (c)
 ΔH = the terminal pressure (head) loss for the bed, ft.
 k_i = a function of the geometric mean grain diameter d_{gi} in layer i . For rounded grains, the Kozeny–Carmen equation can be used to estimate k_i :
 $k_i = 0.081 d_{gi}^{-2}$, where d_{gi} is in millimeters.

Typical values are $n = 1$, $d_g = 1$ mm, $\Delta H = 10$ ft, $D = 3$ ft, and $(K_\rho/\beta C_0) = 700$.
 The backwash flow rate is calculated from

$$q_b = \left(\frac{V_f}{V_f - V_b} - 1 \right) q \quad (d)$$

We assume the backwash water is not recycled.

We next summarize the annual operating costs of the filter because they are equal to the energy costs for pumping

$$f_0 = q_b \left[1.146 \times 10^{-3} C_E \left(\frac{h}{\eta} \right) \right] \quad (e)$$

- where f_0 = dollars per year
 h = the backwash pumping head in feet of water

C_E = the cost of electricity in dollars per kilowatt-hour

η = the pump efficiency

1.146×10^{-3} = the conversion factor

Let us now carry out a numerical calculation based on the following values for the filter parameters

$$h = 110 \text{ ft of water (33.5 m)}$$

$$\eta = 0.8$$

$$b = \frac{\$870}{(\text{ft}^2)^{0.86}}, \frac{\$6715}{(\text{m}^2)^{0.86}}$$

$$z = 0.86$$

$$r = 0.134 \text{ (12.5\% for 20 years) (year}^{-1}\text{)}$$

$$C_E = 0.03/\text{kWh}$$

Substitution of these values into Equations (a) and (e) together with Equations (b) and (d) yields the total cost function

$$f\left(\frac{\$}{\text{year}}\right) = 116 \left[\frac{10^6 q}{1440 / [(V_f/Q) + t_b](V_f - V_b)} \right]_{0 \leq q \leq 10}^{0.86} + 4.73 \times 10^3 \left[\frac{V_f}{V_f - V_b} - 1 \right] q \quad (f)$$

If the values of q , t_b , and V_b are specified, and Equation (c) is ignored, the total annual cost can be determined as a function of the water production V_f per bed area and the filtration rate Q .

Figure E13.3 shows f versus V_f , the water filtered per run, for q (in 10^6 units) = 10 Mgal/day (3.79×10 ML/day), $t_b = 10$ min, and $V_b = 200$ gal/ft² (8.15×10^3 L/m²)

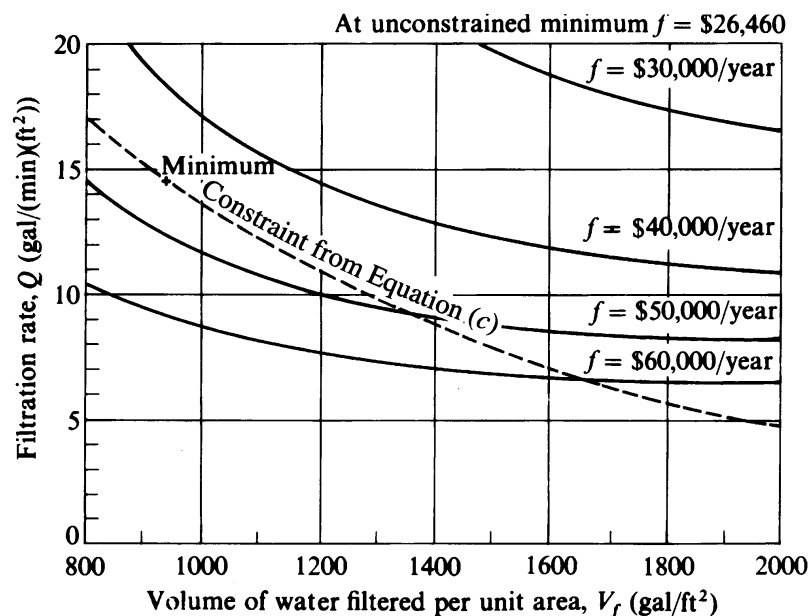


FIGURE E13.3

with Q gal/(min)(ft²) as a parameter. The unconstrained solution is at the upper bounds on Q and V_f . Notice the flatness of f as V_f increases.

Equation (c) would be used in the design of the filter, hence Equation (c) imposes a constraint that must be taken into account. The optimal solution becomes $V_f = 940$ gal/ft² and $Q = 14.2$ gal/(min)(ft²) with Equation (c) included in the problem (see Figure E13.3). A rule of thumb is 2 gal/(min)(ft²) (Letterman, 1980), as compared with the optimal value of Q .

EXAMPLE 13.4 OPTIMAL DESIGN OF A GAS TRANSMISSION NETWORK

A gas-gathering and transmission system consists of sources of gas, arcs composed of pipeline segments, compressor stations, and delivery sites. The design or expansion of a gas pipeline transmission system involves capital expenditures as well as the continuing cost of operation and maintenance. Many factors have to be considered, including

1. The maximum number of compressor stations that would ever be required during a specified time horizon
2. The optimal locations of these compressor stations
3. The initial construction dates of the stations
4. The optimal solution for the expansion for the compressor stations
5. The optimal diameter sizes of the main pipes for each arc of the network
6. The minimum recommended thickness of the main pipes
7. The optimal diameter sizes, thicknesses, and lengths of any required parallel pipe loops on each arc of the network
8. The timing of constructions of the parallel pipe loops
9. The operating pressures of the compressors and the gas in the pipelines

In this example we describe the solution of a simplified problem so that the various factors involved are clear. Suppose that a gas pipeline is to be designed so that it transports a prespecified quantity of gas per time from point A to other points. Both the initial state (pressure, temperature, composition) at A and final states of the gas are known. We need to determine.

1. The number of compressor stations
2. The lengths of pipeline segments between compressor stations
3. The diameters of the pipeline segments
4. The suction and discharge pressures at each station.

The criterion for the design will be the minimum total cost of operation per year including capital, operation, and maintenance costs. Note that the problem considered here does *not* fix the number of compressor stations, the pipeline lengths, the diameters of pipe between stations, the location of branching points, nor limit the configuration (branches) of the system so that the design problem has to be formulated as a nonlinear integer programming problem. Figure E13.4a illustrates a simplified pipeline that we use in defining and solving the problem.

Before presenting the details of the design problem, we need to distinguish between two related problems, one being of a higher degree of difficulty than the other. If the capital costs of the compressors are a linear function of horsepower as shown in line A in Figure E13.4b, the transmission line problem can be solved as a nonlinear programming problem by one of the methods discussed in Chapter 8. On the other

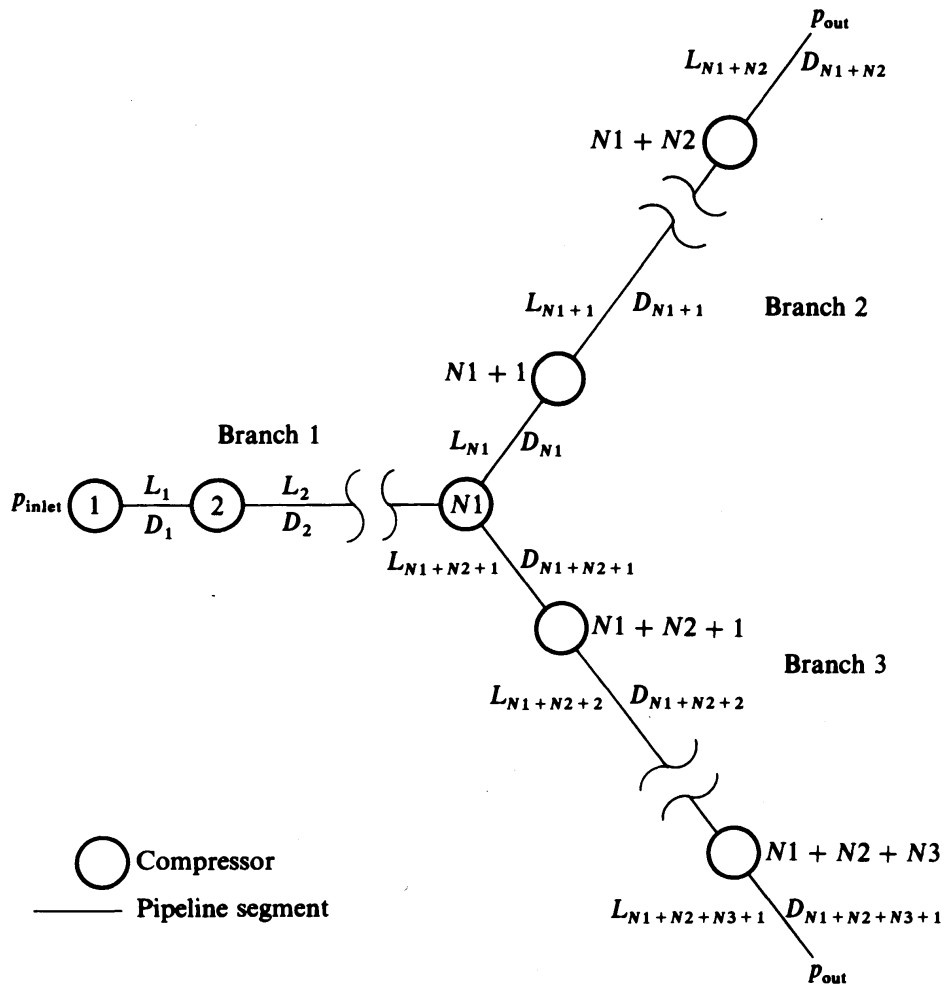


FIGURE E13.4a
Pipeline configuration with three branches.

hand, if the capital costs are a linear function of horsepower with a fixed capital outlay for zero horsepower as indicated by line B in Figure E13.4b, a condition that more properly reflects the real world, then the design problem becomes more difficult to solve and must be solved by a branch-and-bound algorithm combined with a nonlinear programming algorithm as discussed later on. The reason why the branch-and-bound method is avoided for the case involving line A is best examined after the mathematical formulation of the objective function (cost function) has been completed. We split the discussion of the transmission line problem into five parts: (1) the pipeline configuration, (2) the variables, (3) the objective function and costs, (4) the inequality constraints, and (5) the equality constraints.

The pipeline configuration. Figure E13.4a shows the configuration of the pipeline we are using in this example and the notation employed for the numbering system for the compressor stations and the pipeline segments. Each compressor station is represented by a node and each pipeline segment by an arc. $N1$, $N2$, and $N3$ represent the maximum number of possible stations in each of the three branches. Pressure increases at a compressor and decreases along the pipeline segment. The transmission

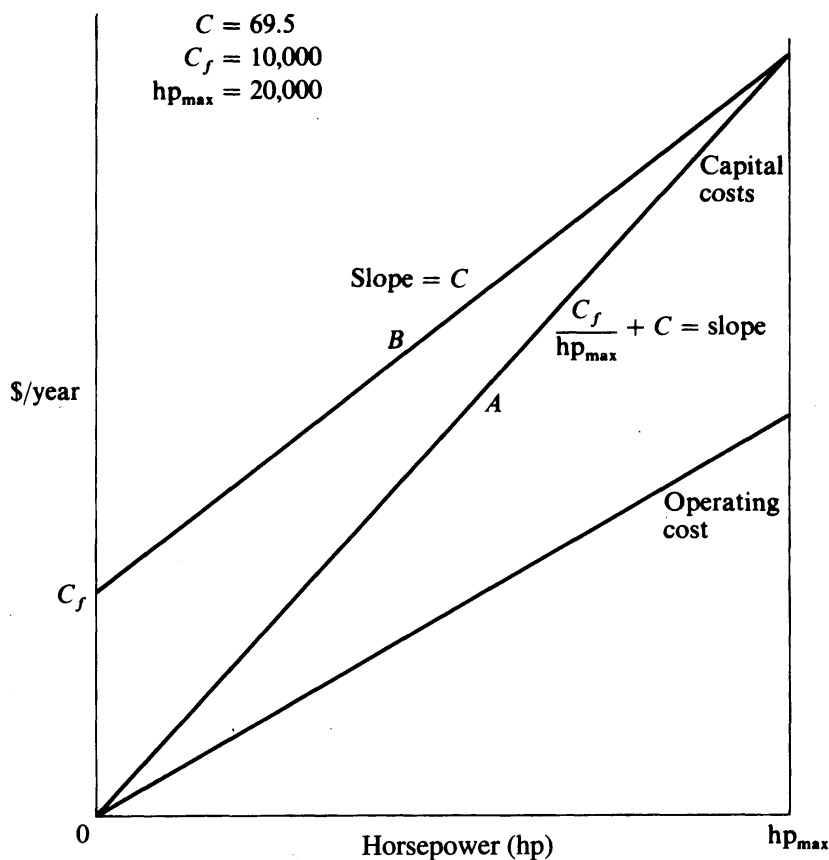


FIGURE E13.4b
Capital and operating costs of compressors.

system is presumed to be horizontal. Although a simple example has been selected to illustrate a transmission system, a much more complicated network can be accommodated that includes various branches and loops at the cost of additional computation time. For a given pipeline configuration each node and each arc are labeled separately. In total there are

- n total compressors [$n = \sum (N_i)$]
- $n - 1$ suction pressures (the initial entering pressure is known)
- n discharge pressures
- $n + 1$ pipeline segment lengths and diameters (note there are two segments issuing at the branch)

The variables. Each pipeline segment has associated with it five variables: (1) the flow rate Q ; (2) the inlet pressure p_d (discharge pressure from the upstream compressor); (3) the outlet pressure p_s (suction pressure of the downstream compressor), (4) the pipe diameter D , and (5) the pipeline segment length L . Inasmuch as the mass flow rate is fixed, and each compressor is assumed to have gas consumed for operation of one-half of one percent of the gas transmitted, only the last four variables need to be determined for each segment.

The objective function. Because the problem is posed as a minimum cost problem, the objective function is the sum of the yearly operating and maintenance

costs of the compressors plus the sum of the discounted (over 10 years) capital costs of the pipeline segments and compressors. Each compressor is assumed to be adiabatic with an inlet temperature equal to that of the surroundings. A long pipeline segment is assumed so that by the time gas reaches the next compressor it returns to the ambient temperature. The annualized capital costs for each pipeline segment depend on pipe diameter and length, but are assumed to be \$870/(in.)(mile)(year). The rate of work of one compressor is

$$W = (0.08531)Q \frac{k}{k-1} T_1 \left[\left(\frac{p_d}{p_s} \right)^{z(k-1)/k} - 1 \right] \quad (a)$$

where $k = C_p/C_v$ for gas at suction conditions (assumed to be 1.26)

z = compressibility factor of gas at suction conditions (z ranges from 0.88 to 0.92)

p_s = suction pressure, psi

p_d = discharge pressure, psi

T_1 = suction temperature, °R (assumed 520°R)

Q = flow rate into the compressor, MMCFD (million cubic feet per day)

W = rate of work, horsepower.

Operation and maintenance charges per year can be related directly to horsepower and are estimated to be between 8.00 and 14.0 \$(/hp)(year), hence the total operating costs are assumed to be a linear function of compressor horsepower.

Figure E13.4b shows two different forms for the annualized capital cost of the compressors. Line A indicates the cost is a linear function of horsepower [\$70.00/(hp)(year)] with the line passing through the origin, whereas line B assumes a linear function of horsepower with a fixed initial capital outlay [\$70.00/(hp)(year) + \$10,000] to take into account installation costs, foundation, and so on. For line A, the objective function in dollars per year for the example problem is

$$f = \sum_{i=1}^n (C_0 + C_c) Q_i (0.08531) T_1 \left(\frac{k}{k-1} \right) \left[\left(\frac{p_{d_i}}{p_{s_i}} \right)^{z(k-1)/k} - 1 \right] + \sum_{j=1}^m C_s L_j D_j \quad (b)$$

where n = number of compressors in the system

m = number of pipeline segments in the system ($= n + 1$)

C_0 = yearly operating cost \$(/hp)(year)

C_c = compressor capital cost \$(/hp)(year)

C_s = pipe capital cost \$(/in.)(mile)(year)

L_j = length of pipeline segment j , mile

D_j = diameter of pipeline segment j , in.

You can now see why for line A a branch-and-bound technique is not required to solve the design problem. Because of the way the objective function is formulated, if the ratio $(p_d/p_s) = 1$, the term involving compressor i vanishes from the first summation in the objective function. This outcome is equivalent to the deletion of compressor i in the execution of a branch-and-bound strategy. (Of course the pipeline segments joined at node i may be of different diameters.) But when

line B represents the compressor costs, the fixed incremental cost for each compressor in the system at zero horsepower (C_f) is *not* multiplied by the term in the square brackets of Equation (b). Instead, C_f is added in the sum of the costs whether or not compressor i is in the system, and a nonlinear programming technique cannot be used alone. Hence, if line B applies, a different solution procedure is required.

The inequality constraints. The operation of each compressor is constrained so that the discharge pressure is greater than or equal to the suction pressure

$$\frac{p_{d_i}}{p_{s_i}} \leq 1, \quad i = 1, 2, \dots, n \quad (c)$$

and the compression ratio does not exceed some prespecified maximum limit K

$$\frac{p_{d_i}}{p_{s_i}} \geq K_i, \quad i = 1, 2, \dots, n \quad (d)$$

In addition, upper and lower bounds are placed on each of the four variables

$$p_{d_i}^{\min} \leq p_{d_i} \leq p_{d_i}^{\max} \quad (e)$$

$$p_{s_i}^{\min} \leq p_{s_i} \leq p_{s_i}^{\max} \quad (f)$$

$$L_i^{\min} \leq L_i \leq L_i^{\max} \quad (g)$$

$$D_i^{\min} \leq D_i \leq D_i^{\max} \quad (h)$$

The equality constraints. Two classes of equality constraints exist for the transmission system. First, the length of the system is fixed. With two branches, there are two constraints

$$\sum_{j=1}^{N1-1} L_j + \sum_{j=N1}^{N1+N2} L_j = L_1^*$$

$$\sum_{j=1}^{N1-1} L_j + \sum_{j=N1+N2+1}^{1N+N2+N3+1} L_j = L_2^* \quad (i)$$

where L_k^* represents the length of a branch. Second, the flow equation, the Weymouth relation (GPSA handbook, 1972), must hold in each pipeline segment

$$Q_j = 871D_j^{8/3} \left[\frac{p_d^2 - p_s^2}{L_j} \right]^{1/2} \quad (j)$$

where $Q_j =$ a fixed number

$p_d =$ the discharge pressure at the entrance of the segment

$p_s =$ the suction pressure at the exit of the segment

To avoid problems in taking square roots, Equation (j) is squared to yield

$$(871)^2 D_j^{16/3} (p_d^2 - p_s^2) - L_j Q_j^2 = 0 \quad (k)$$

Solution strategy. As mentioned previously, if the capital costs in the problem are described by line *A* in Figure E13.4b, then the problem can be solved directly by a nonlinear programming algorithm. If the capital costs are represented by line *B* in Figure E13.4b, then nonlinear programming in conjunction with branch-and-bound enumeration must be used to accommodate the integer variable of a compressor being in place or not.

As explained in Chapter 9, a branch-and-bound enumeration is nothing more than a search organized so that certain portions of the possible solution set are deleted from consideration. A tree is formed of nodes and branches (arcs). Each branch in the tree represents an added or modified inequality constraint to the problem defined for the prior node. Each node of the tree itself represents a nonlinear optimization problem without integer variables.

With respect to the example we are considering, in Figure E13.4c, node 1 in the tree represents the original problem as posed by Equations (b)–(k), that is the problem in which the capital costs are represented by line *A* in Figure E13.4b. When the problem at node 1 is solved, it provides a lower bound on the solution of the problem involving the cost function represented by line *B* in Figure E13.4b. Note that line *A* always lies below line *B*. (If the problem at node 1 using line *A* had no feasible solution, the more complex problem involving line *B* also would have no feasible solution.) Although the solution of the problem at node 1 is feasible, the solution may not be feasible for the problem defined by line *B* because line *B* involves an initial fixed capital cost at zero horsepower.

After solving the problem at node 1, a decision is made to partition on one of the three integer variables; N_1 , N_2 , or N_3 . The partition variable is determined by the following heuristic rule.

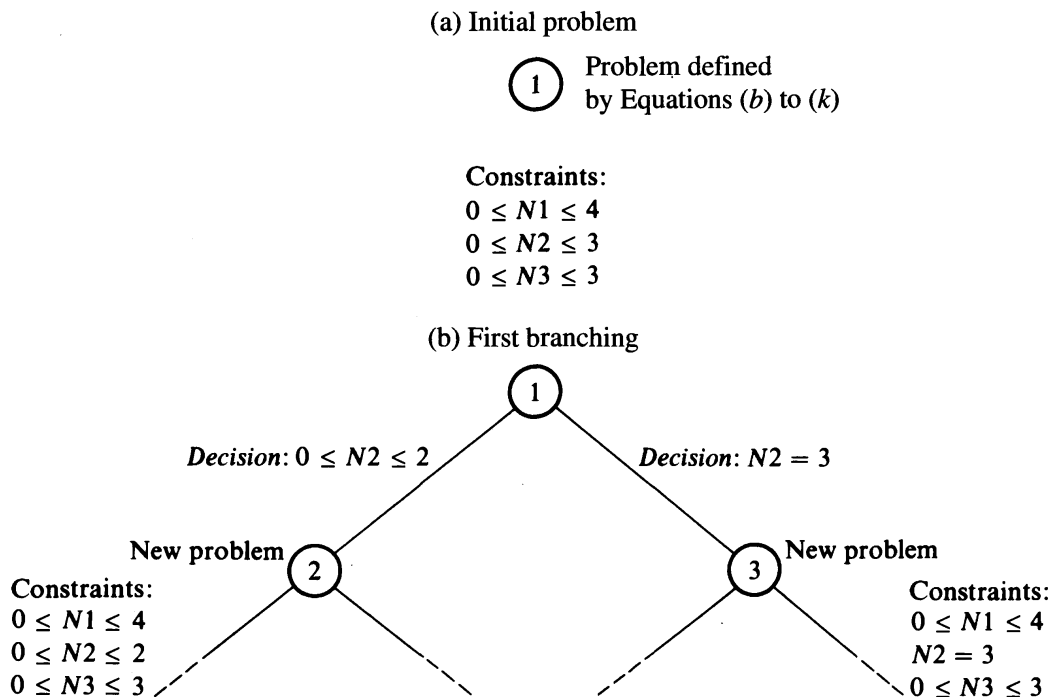


FIGURE E13.4c

Partial tree and branches for the example design problem.

The smallest average compression ratio of all the branches in the transmission system is calculated by adding all the compression ratios in each branch and dividing by the number of compressors in the branch. The number of compressors in the branch that has smallest ratio becomes the partition variable.

Based on this rule, the partition variable was calculated to be N_2 .

After selection of the partition variable, the next step is to determine how the variable should be partitioned. It was decided to check each compressor in the branch of the transmission line associated with the partition variable, and if any compressor operated at less than 10 percent of capacity, it was assumed the compressor was not necessary in the line. (If all operate at greater than 10 percent capacity, the compressor with the smallest compression ratio was deleted.) For example, with N_2 selected as the partition variable, and one of the three possible compressors in branch 2 of the gas transmission network operating at less than 10 percent of capacity, the first partition would lead to the tree shown in Figure E13.4c; N_2 would either be 3 or would be $0 \leq N_2 \leq 2$. Thus at each node in the tree, the upper or lower bound on the number of compressors in each branch of the pipeline is readjusted to be tighter.

The nonlinear problem at node 2 is the same as at node 1, with two exceptions. First, the maximum number of compressors permitted in branch 2 of the transmission line is now two. Second, the objective function is changed. From the lower bounds, we know the minimum number of compressors in each branch of the pipeline. For the lower bound, the costs related to line *B* in Figure E13.4b apply; for compressors in excess of the lower bound and up to the upper bound, the costs are represented by line *A*.

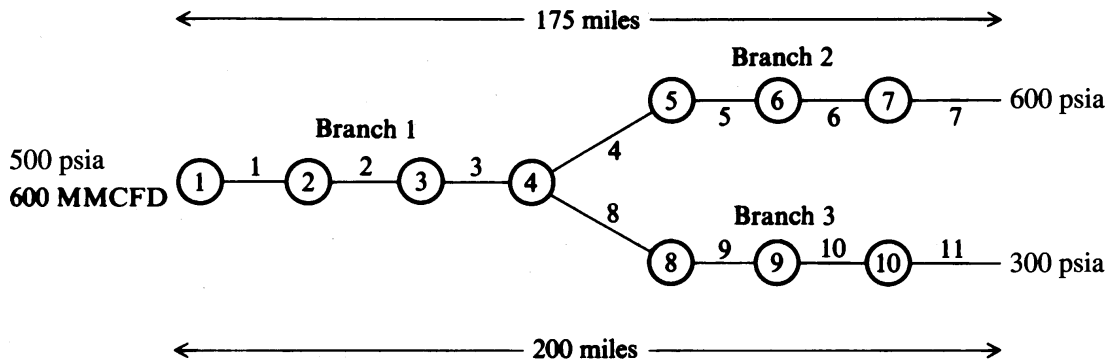
As the decision tree descends, the solution at each node becomes more and more constrained, until node r is reached, in which the upper bound and the lower bound for the number of compressors in each pipeline branch are the same. The solution at node r is feasible for the general problem but not necessarily optimal. Nevertheless, the important point is that the solution at node r is an upper bound on the solution of the general problem.

As the search continues through the rest of the tree, if the value of the objective function at a node is greater than that of the best feasible solution found to that stage in the search, then it is not necessary to continue down that branch of the tree. The objective function of any solution subsequently found in the branch is larger than the solution already found. Thus, we can fathom the node, that is, terminate the search down that branch of the tree.

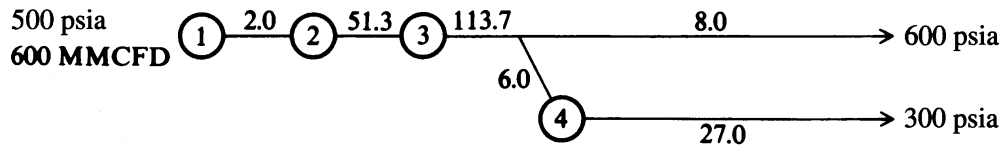
The next step is to backtrack up the tree and continue searching through other branches until all nodes in the tree have been fathomed. Another reason to fathom a particular node occurs when no feasible solution exists to the nonlinear problem at node r ; then all subsequent nodes below node r are also infeasible.

At the end of the search, the best solution found is the solution to the general problem.

Computational results. Figure E13.4d and Table E13.4A show the solution to the example design problem outlined in Figure E13.4a using the cost relation of line *A* in Figure E13.4b. The maximum number of compressors in branches 1, 2, and 3 were set at 4, 3, and 3, respectively. The input pressure was fixed at 500 psi at a flow rate of 600 MMCFD, and the two output pressures were set at 600 psi and 300 psi, respectively, for branches 2 and 3. The total length of branches 1 plus 2 was constrained to be 175 miles, whereas the total length of branches 1 plus 3 was constrained



(a) Initial configuration (arcs are numbered sequentially)



(b) Optimal configuration with optimal pipeline lengths (in miles) shown on the arcs

FIGURE E13.4d

Initial gas transmission system and final optimal system using the costs of line A, Figure E13.4b.

at 200 miles. The upper bound on the diameter of the pipeline segments in branch 1 was set at 36 inches, the upper bound on the diameters of the pipeline segments in branches 2 and 3 at 18 in., and the lower bound on the diameters of all pipeline segments at 4 in. A lower bound of 2 miles was placed on each pipeline segment to ensure that the natural gas was at ambient conditions when it entered a subsequent compressor in the pipeline.

Figure E13.4d compares the optimal gas transmission network with the original network. From a nonfeasible starting configuration with 10-mile-long pipeline segments, the nonlinear optimization algorithm reduced the objective function from the first feasible state of 1.399×10^7 dollars/year to 7.289×10^6 dollars/year, a savings of close to \$7 million. Of the ten possible compressor stations, only four remained in the final optimal network. Table E13.4a lists the final state of the network. Note that because the suction and discharge pressures for the pipeline segments in branch 2 are identical, compressors 4, 5, 6, and 7 do not exist in the optimal configuration, nor do 9 and 10 in branch 3.

The same problem represented by Figure E13.4a was solved again but using the costs represented by line B instead of line A in Figure E13.4b. Figure E13.4e and Table E13.4B present the results of the computations. It is interesting to note that compressor 3 remains in the final configuration but with a compression ratio of 1, that is, compressor 3 is not doing any work. This means that it is cheaper to have two

TABLE E13.4A
Values of operating variables for the optimal network configuration
using the costs of line A, Figure E13.4b

Pipeline segment	Discharge pressure (psi)	Suction pressure (psi)	Pipe diameter (in.)	Length (mile)	Flow rate (MMCFD)
1	719.1	715.4	35.0	2.0	597.0
2	1000.0	889.3	32.4	51.3	594.0
3	1000.0	735.8	32.4	113.7	591.0
4	735.7	703.8	18.0	2.0	294.0
5	703.8	670.6	18.0	2.0	292.6
6	670.6	636.1	18.0	2.0	291.1
7	636.1	600.0	18.0	2.0	289.7
8	735.8	703.8	18.0	2.0	294.0
9	685.2	859.1	18.0	2.0	292.6
10	859.1	832.5	18.0	2.0	291.1
11	832.5	300.0	18.0	27.0	289.7

Compressor station	Compression ratio	Capital cost (\$/year)
1	1.44	70.00
2	1.40	70.00
3	1.00	70.00
4	1.00	70.00
5	1.00	70.00
6	1.00	70.00
7	1.00	70.00
8	1.26	70.00
9	1.00	70.00
10	1.00	70.00

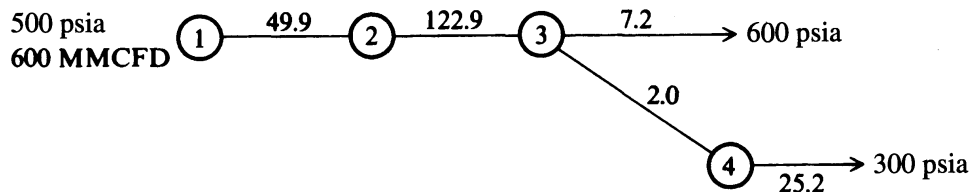


FIGURE E13.4e

Optimal configuration using the costs of line B in Figure E13.4b.

pipeline segments in branch 1 and two compressors each operating at about one-half capacity, plus a penalty of \$10,000, than to have one pipeline segment and one compressor operating at full capacity. Compressor 3 doing no work represents just a branch in the line plus a cost penalty.

TABLE E13.4B
Values of operating variables for the optimal network configuration
using the costs of line B, Figure E13.4b

Pipeline segment	Discharge pressure (psi)	Suction pressure (psi)	Pipe diameter (in.)	Length (mile)	Flow rate (MMCFD)
1	954.5	837.2	32.3	49.9	597.0
2	1000.0	699.7	32.3	122.9	594.0
3	699.7	600.0	15.2	2.2	295.5
4	699.7	665.7	18.0	2.0	295.5
5	952.2	300.0	16.9	25.2	294.0

Compressor station	Compression ratio	Capital cost (\$/year)
1	1.91	69.50
2	1.19	69.50
3	1.00	69.50
4	1.43	69.50

REFERENCES

- Gas Processor Suppliers Association. *Engineering Data Book*. 1972.
- Letterman, R. D. "Economic Analysis of Granular Bed Filtration." *Trans Am Soc Civil Engrs* **106**: 279 (1980).
- McCabe, W. L.; J. C. Smith; and P. Harriott. *Unit Operations of Chemical Engineering*, 5th ed. McGraw-Hill, New York (1993).

SUPPLEMENTARY REFERENCES

- Bejan, A. "Maximum Power from Fluid Flow." *Int J Heat Mass Transfer* **39**: 1175–1181 (1996).
- Carroll, J. A.; and R. N. Horne. "Multivariate Optimization of Production Systems." *J Petrol Tech* **44**: 782–789 (July, 1992).
- Currie, J. C.; J. F. Novotrak; B. T. Ashdee; and C. J. Kennedy. "Optimize Reservoir Management: Mixed Linear Programming." *J Petrol Tech* 1351–1355 (December, 1997).
- Heinemann, R. F.; and S. L. Lyons. "Next Generation Reservoir Optimization." *World Oil* **219**: 47–50 (1998).
- Nishikiori, N.; R. A. Redner; and D. R. Doty. "An Improved Method for Gas Lift Allocation Optimization." *J Energy Resour Tech* **117**: 87–92 (1995).

- Pan, Y.; and R. N. Horne. "Multivariate Optimization of Field Development Scheduling and Well-Placement Design." *J Petrol Tech* 83–86 (December, 1998).
- Ramirez, W. F. *Application of Optimal Control to Enhanced Oil Recovery*. Elsevier, Amsterdam (1998).
- Sung, W.; D. Huh; and J. Lee. "Optimization of Pipeline Networks with a Hybrid MCSTCD Networking Model." *SPE Prod Fac* 13(3): 213–219 (1998).

CHEMICAL REACTOR DESIGN AND OPERATION

Example

14.1 Optimization of a Thermal Cracker Via Linear Programming	484
14.2 Optimal Design of an Ammonia Reactor	488
14.3 Solution of an Alkylation Process by Sequential Quadratic Programming	492
14.4 Predicting Protein Folding	495
14.5 Optimization of Low-Pressure Chemical Vapor Deposition Reactor for the Deposition of Thin Films	500
14.6 Reaction Synthesis Via MINLP	508
References	513
Supplementary References	514

IN PRACTICE, EVERY chemical reaction carried out on a commercial scale involves the transfer of reactants and products of reaction, and the absorption or evolution of heat. Physical design of the reactor depends on the required structure and dimensions of the reactor, which must take into account the temperature and pressure distribution and the rate of chemical reaction. In this chapter, after describing the methods of formulating optimization problems for reactors and the tools for their solution, we will illustrate the techniques involved for several different processes.

Modeling chemical reactors

Optimization in the design and operation of a reactor focuses on formulating a suitable objective function plus a mathematical description of the reactor; the latter forms a set of constraints. Reactors in chemical engineering are usually, but not always, represented by one or a combination of

1. Algebraic equations
2. Ordinary differential equations
3. Partial differential equations

One extreme of representation of reactor operation is complete mixing in a continuous stirred tank reactor (CSTR); the other extreme is no mixing whatsoever (plug flow). In between are various degrees of mixing within dispersion reactors. Single ideal reactor types can be combined in various configurations to represent intermediate types of mixing as well as nonideal mixing and fluid bypassing.

Ideal reactors can be classified in various ways, but for our purposes the most convenient method uses the mathematical description of the reactor, as listed in Table 14.1. Each of the reactor types in Table 14.1 can be expressed in terms of integral equations, differential equations, or difference equations. Not all real reactors can fit neatly into the classification in Table 14.1, however. The accuracy and precision of the mathematical description rest not only on the character of the mixing and the heat and mass transfer coefficients in the reactor, but also on the validity and analysis of the experimental data used to model the chemical reactions involved.

Other factors that must be considered in the modeling of reactors, factors that influence the number of equations and their degree of nonlinearity but not their form, are

1. The number and nature of the phases present in the reactor (gas, liquid, solid, and combinations thereof)
2. The method of supplying and removing heat (adiabatic, heat exchange mechanism, etc.)
3. The geometric configuration (empty cylinder, packed bed, sphere, etc.)
4. Reaction features (exothermic, endothermic, reversible, irreversible, number of species, parallel, consecutive, chain, selectivity)
5. Stability
6. The catalyst characteristics

Some references for the modeling of chemical reactors include Fogler (1998), Froment and Bischoff (1990), Levenspiel (1998), Missen and colleagues, (1998), and Schmidt (1997).

TABLE 14.1
Classification of reactors

Reactor type	Mathematical description (continuous variables)
Batch [well-mixed (CSTR), closed system]	Ordinary differential equations (unsteady state) Algebraic equation (steady state)
Semibatch [well-mixed (CSTR), open system]	Ordinary differential equations (unsteady state) Algebraic equations (steady state)
CSTRs, individual or in series	Ordinary differential equations (unsteady state) Algebraic equations (steady state)
Plug flow reactor	Partial differential equations in one spatial variable (unsteady state) Ordinary differential equations in the spatial variable (steady state)
Dispersion reactor	Partial differential equations (unsteady state and steady state) Ordinary differential equations in one spatial variable (steady state)

Abbreviation: CSTR = continuous stirred tank reactor.

Objective functions for reactors

Various questions that lead directly to the formulation of an objective function can be posed concerning reactors. Typical objective functions stated in terms of the adjustable variables are

1. Maximize conversion (yield) per volume with respect to time.
2. Maximize production per batch.
3. Minimize production time for a fixed yield.
4. Minimize total production costs per average production costs with respect to time per fraction conversion.
5. Maximize yield per number of moles of component per concentration with respect to time or operating conditions.
6. Design the optimal temperature sequence with respect to time per reactor length to obtain (a) a given fraction conversion, (b) a maximum rate of reaction, or (c) the minimum residence time.
7. Adjust the temperature profile to specifications (via sum of squares) with respect to the independent variables.
8. Minimize volume of the reactor(s) with respect to certain concentration(s).
9. Change the temperature from T_0 to T_f in minimum time subject to heat transfer rate constraints.
10. Maximize profit with respect to volume.

11. Maximize profit with respect to fraction conversion to get optimal recycle.
12. Optimize profit per volume per yield with respect to boundary per initial conditions in time.
13. Minimize consumption of energy with respect to operating conditions.

In some cases a variable can be independent and in others the same variable can be dependent, but the usual independent variables are pressure, temperature, and flow rate or concentration of a feed. We cannot provide examples for all of these criteria, but have selected a few to show how they mesh with the optimization methods described in earlier chapters and mathematical models listed in Table 14.1.

In considering a reactor by itself, as we do in this chapter, keep in mind that a reactor will no doubt be only one unit in a complete process, and that at least a separator must be included in any economic analysis. Figure 14.1 depicts the relation between the yield or selectivity of a reactor and costs.

All of the various optimization techniques described in previous chapters can be applied to one or more types of reactor models. The reactor model forms a set of constraints so that most optimization problems involving reactors must accommodate steady-state algebraic equations or dynamic differential equations as well as inequality constraints.

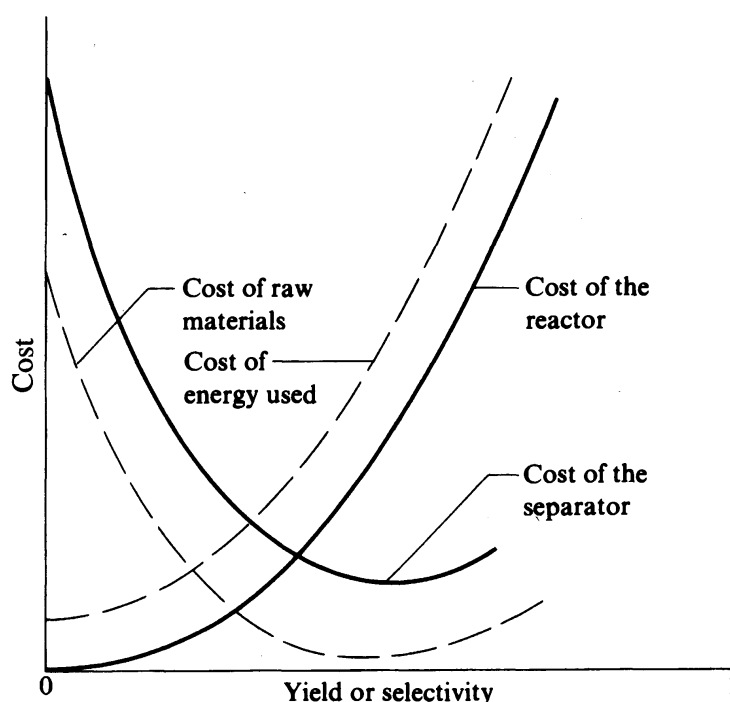


FIGURE 14.1

Costs of energy and raw materials for a reactor as a function of yield and selectivity [Adapted and modified from P. LeGoff, "The Energetic and Economic Optimization of Heterogeneous Reactors," *Chem Eng Sci* **35**: 2089 (1980)].

EXAMPLE 14.1 OPTIMIZATION OF A THERMAL CRACKER VIA LINEAR PROGRAMMING

Reactor systems that can be described by a “yield matrix” are potential candidates for the application of linear programming. In these situations, each reactant is known to produce a certain distribution of products. When multiple reactants are employed, it is desirable to optimize the amounts of each reactant so that the products satisfy flow and demand constraints. Linear programming has become widely adopted in scheduling production in olefin units and catalytic crackers. In this example, we illustrate the use of linear programming to optimize the operation of a thermal cracker sketched in Figure E14.1.

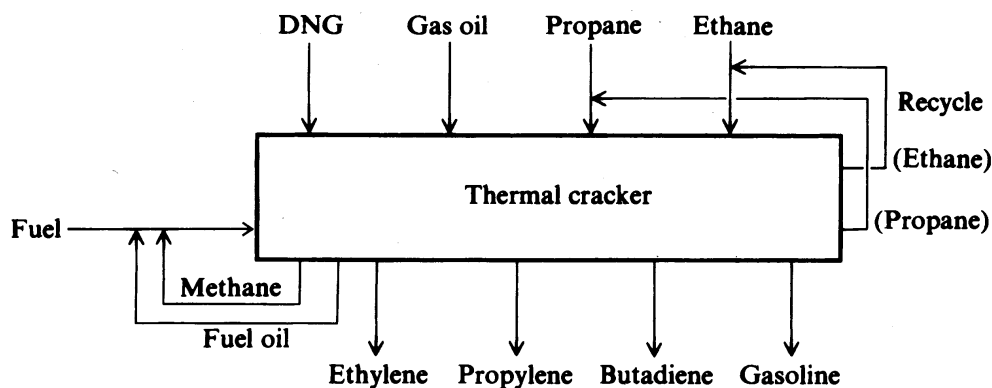


FIGURE E14.1
Flow diagram of thermal cracker.

Table E14.1A shows various feeds and the corresponding product distribution for a thermal cracker that produces olefins. The possible feeds include ethane, propane, debutanized natural gasoline (DNG), and gas oil, some of which may be fed simultaneously. Based on plant data, eight products are produced in varying proportions according to the following matrix. The capacity to run gas feeds through the cracker is 200,000 lb/stream hour (total flow based on an average mixture). Ethane uses the equivalent of 1.1 lb of capacity per pound of ethane; propane 0.9 lb; gas oil 0.9 lb/lb; and DNG 1.0.

TABLE E14.1A
Yield structure: (wt. fraction)

Product	Feed			
	Ethane	Propane	Gas oil	DNG
Methane	0.07	0.25	0.10	0.15
Ethane	0.40	0.06	0.04	0.05
Ethylene	0.50	0.35	0.20	0.25
Propane	—	0.10	0.01	0.01
Propylene	0.01	0.15	0.15	0.18
Butadiene	0.01	0.02	0.04	0.05
Gasoline	0.01	0.07	0.25	0.30
Fuel oil	—	—	0.21	0.01

Downstream processing limits exist of 50,000 lb/stream hour on the ethylene and 20,000 lb/stream hour on the propylene. The fuel requirements to run the cracking system for each feedstock type are as follows:

Feedstock type	Fuel requirement (Btu/lb)
Ethane	8364
Propane	5016
Gas oil	3900
DNG	4553

Methane and fuel oil produced by the cracker are recycled as fuel. All the ethane and propane produced is recycled as feed. Heating values are as follows:

Recycled feed	Heat produced (Btu/lb)
Natural gas	21,520
Methane	21,520
Fuel oil	18,000

Because of heat losses and the energy requirements for pyrolysis, the fixed fuel requirement is 20.0×10^6 Btu/stream hour. The price structure on the feeds and products and fuel costs is:

Feeds	Price (¢/lb)
Ethane	6.55
Propane	9.73
Gas oil	12.50
DNG	10.14

Products	Price (¢/lb)
Methane	5.38 (fuel value)
Ethylene	17.75
Propylene	13.79
Butadiene	26.64
Gasoline	9.93
Fuel oil	4.50 (fuel value)

Assume an energy (fuel) cost of $\$2.50/10^6$ Btu.

The procedure is to

1. Set up the objective function and constraints to maximize profit while operating within furnace and downstream process equipment constraints. The variables to be optimized are the amounts of the four feeds.
2. Solve using linear programming.
3. Examine the sensitivity of profits to increases in the ethylene production rate.

We define the following variables for the flow rates to and from the furnace (in lb/h):

$$x_1 = \text{fresh ethane feed}$$

$$x_2 = \text{fresh propane feed}$$

x_3 = gas oil feed

x_4 = DNG feed

x_5 = ethane recycle

x_6 = propane recycle

x_7 = fuel added

Assumptions used in formulating the objective function and constraints are

1. 20×10^6 Btu/h fixed fuel requirement (methane) to compensate for the heat loss.
2. All propane and ethane are recycled with the feed, and all methane and fuel oil are recycled as fuel.

A basis of 1 hour is used, and all costs are calculated in cents per hour.

Objective function (profit). In words, the profit f is

$$f = \text{Product value} - \text{Feed cost} - \text{Energy cost}$$

Product value. The value for each product (in cents per pound) is as follows:

$$\text{Ethylene: } 17.75(0.5x_1 + 0.5x_5 + 0.35x_2 + 0.35x_6 + 0.20x_3 + 0.25x_4) \quad (a)$$

$$\text{Propylene: } 13.79(0.01x_1 + 0.01x_5 + 0.15x_2 + 0.15x_6 + 0.15x_3 + 0.18x_4) \quad (b)$$

$$\text{Butadiene: } 26.64(0.01x_1 + 0.01x_5 + 0.02x_2 + 0.02x_6 + 0.04x_3 + 0.05x_4) \quad (c)$$

$$\text{Gasoline: } 9.93(0.01x_1 + 0.01x_5 + 0.07x_2 + 0.07x_6 + 0.25x_3 + 0.30x_4) \quad (d)$$

$$\text{Total product sales} = 9.39x_1 + 9.51x_2 + 9.17x_3 + 11.23x_4 + 9.39x_5 + 9.51x_6 \quad (e)$$

Feed cost.

$$\text{Feed cost } (\$/h) = 6.55x_1 + 9.73x_2 + 12.50x_3 + 10.14x_4 \quad (f)$$

Energy cost. The fixed heat loss of 20×10^6 Btu/h can be expressed in terms of methane cost (5.38¢/lb) using a heating value of 21,520 Btu/lb for methane. The fixed heat loss represents a constant cost that is independent of the variables x_i , hence in optimization we can ignore this factor, but in evaluating the final costs this term must be taken into account. The value for x_7 depends on the amount of fuel oil and methane produced in the cracker (x_7 provides for any deficit in products recycled as fuel).

We combine (e) and (f) to get the objective function (¢/h)

$$f = 2.84x_1 - 0.22x_2 - 3.33x_3 + 1.09x_4 + 9.39x_5 + 9.51x_6 \quad (g)$$

Constraints.

1. Cracker capacity of 200,000 lb/h

$$1.1(x_1 + x_5) + 0.9(x_2 + x_6) + 0.9x_3 + 1.0x_4 \leq 200,000 \quad (h)$$

or

$$1.1x_1 + 0.9x_2 + 0.9x_3 + 1.0x_4 + 1.1x_5 + 0.9x_6 \leq 200,000$$

2. Ethylene processing limitation of 100,000 lb/h

$$0.5x_1 + 0.35x_2 + 0.25x_3 + 0.25x_4 + 0.5x_5 + 0.35x_6 \leq 100,000 \quad (i)$$

3. Propylene processing limitation of 20,000 lb/h

$$0.01x_1 + 0.15x_2 + 0.15x_3 + 0.18x_4 + 0.01x_5 + 0.15x_6 \leq 20,000 \quad (j)$$

4. Ethane recycle

$$x_5 = 0.4x_1 + 0.4x_5 + 0.06x_2 + 0.06x_6 + 0.04x_3 + 0.05x_4 \quad (k)$$

Rearranging, (j) becomes

$$0.4x_1 + 0.06x_2 + 0.04x_3 + 0.05x_4 - 0.6x_5 + 0.06x_6 = 0 \quad (l)$$

5. Propane recycle

$$x_6 = 0.1x_2 + 0.1x_6 + 0.01x_3 + 0.01x_4 \quad (m)$$

Rearranging Equation (m),

$$0.1x_2 + 0.01x_3 + 0.01x_4 - 0.9x_6 = 0 \quad (n)$$

6. Heat constraint

The total fuel heating value (THV) (in Btu/h) is given by

$$\begin{aligned} \text{THV} &= \begin{array}{c} \text{fuel} \\ 21,520x_7 \end{array} + \begin{array}{c} \text{methane from cracker} \\ 21,520(0.07x_1 + 0.25x_2 + 0.10x_3 + 0.15x_4 - 0.07x_5 + 0.25x_6) \\ \text{fuel oil from cracker} \\ + 18,000(0.21x_3 + 0.01x_4) \end{array} \\ &= 1506.4x_1 + 5380x_2 + 5932x_3 + 3408x_4 + 1506.4x_5 + 5380x_6 + 21,520x_7 \quad (o) \end{aligned}$$

The required fuel for cracking (Btu/h) is

$$\begin{aligned} &\begin{array}{cccc} \text{ethane} & \text{propane} & \text{gas oil} & \text{DNG} \end{array} \\ &8364(x_1 + x_5) + 5016(x_2 + x_6) + 3900x_3 + 4553x_4 \\ &= 8364x_1 + 5016x_2 + 3900x_3 + 4553x_4 + 8364x_5 + 5016x_6 \quad (p) \end{aligned}$$

Therefore the sum of Equation (p) + 20,000,000 Btu/h is equal to the THV from Equation (o), which gives the constraint

$$\begin{aligned} - 6857.6x_1 + 364x_2 + 2032x_3 - 1145x_4 - 6857.6x_5 + 364x_6 \\ + 21,520x_7 = 20,000,000 \quad (q) \end{aligned}$$

Table E14.1B lists the optimal solution of this problem obtained using the Excel Solver (case 1). Note that the maximum amount of ethylene is produced. As the ethylene production constraint is relaxed, the objective function value increases. Once the constraint is raised above 90,909 lb/h, the objective function remains constant.

TABLE E14.1B
Optimal flow rates for cracking furnace for
different restrictions on ethylene and
propylene production

Stream	Flow rate (lb/h)	
	Case 1	Case 2
x_1 (ethane feed)	60,000	21,770
x_2 (propane feed)	0	0
x_3 (gas oil feed)	0	0
x_4 (DNG feed)	0	107,600
x_5 (ethane recycle)	40,000	23,600
x_6 (propane recycle)	0	1,195
x_7 (fuel added)	32,800	21,090
Ethylene	50,000	50,000
Propylene	1,000	20,000
Butadiene	1,000	5,857
Gasoline	1,000	32,820
Methane (recycled to fuel)	7,000	19,610
Fuel oil	0	1,076
Objective function ($\$/h$)	369,560	298,590

Suppose the inequality constraints on ethylene and propylene production were changed to equality constraints (ethylene = 50,000; propylene = 20,000). The optimal solution for these conditions is shown as case 2 in Table E14.1B. This specification forces the use of DNG as well as ethane.

EXAMPLE 14.2 OPTIMAL DESIGN OF AN AMMONIA REACTOR

This example based on the reactor described by Murase et al. (1970) shows one way to mesh the numerical solution of the differential equations in the process model with an optimization code. The reactor, illustrated in Figure E14.2a, is based on the Haber process.

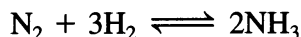


Figure E14.2b illustrates the suboptimal concentration and temperature profiles experienced. The temperature at which the reaction rate is a maximum decreases as the conversion increases.

Assumptions made in developing the model are

1. The rate expression is valid.
2. Longitudinal heat and mass transfer can be ignored.
3. The gas temperature in the catalytic zone is also the catalyst particle temperature.
4. The heat capacities of the reacting gas and feed gas are constant.
5. The catalytic activity is uniform along the reactor and equal to unity.
6. The pressure drop across the reactor is negligible compared with the total pressure in the system.

The notation and data to be used are listed in Table E14.2.

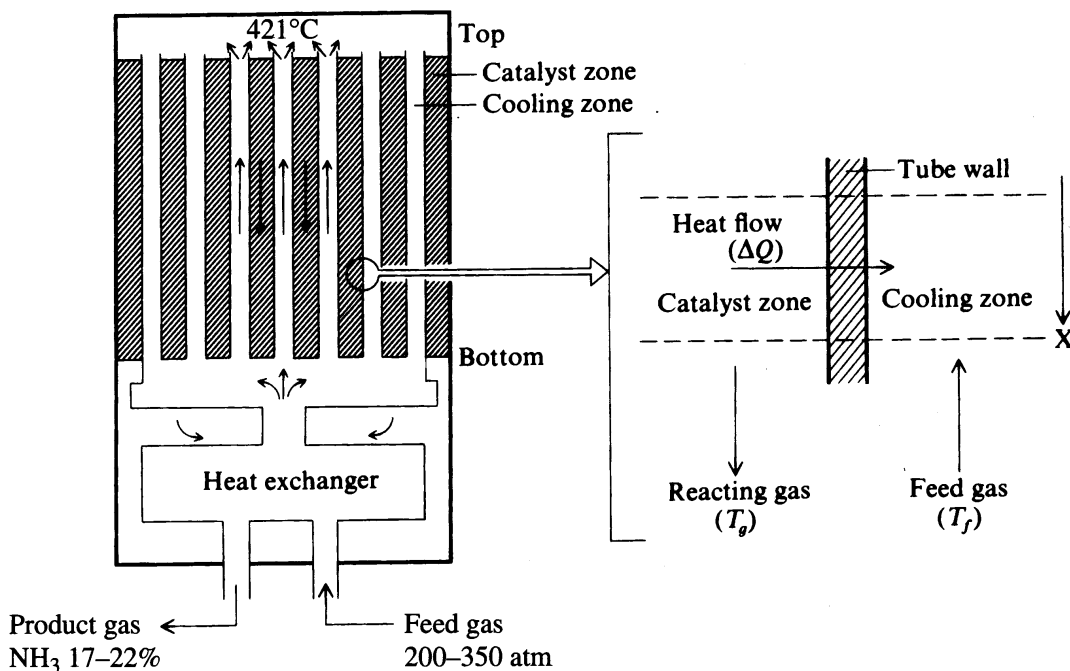


FIGURE E14.2a

Ammonia synthesis reactor. The shaded area contains the catalyst. [Adapted, with permission, from Murase et al., "Optimal Thermal Design of an Auto-thermal Ammonia Synthesis Reactor," *Ind Eng Chem Process Des Dev* **9**: 504 (1970). Copyright, American Chemical Society.]

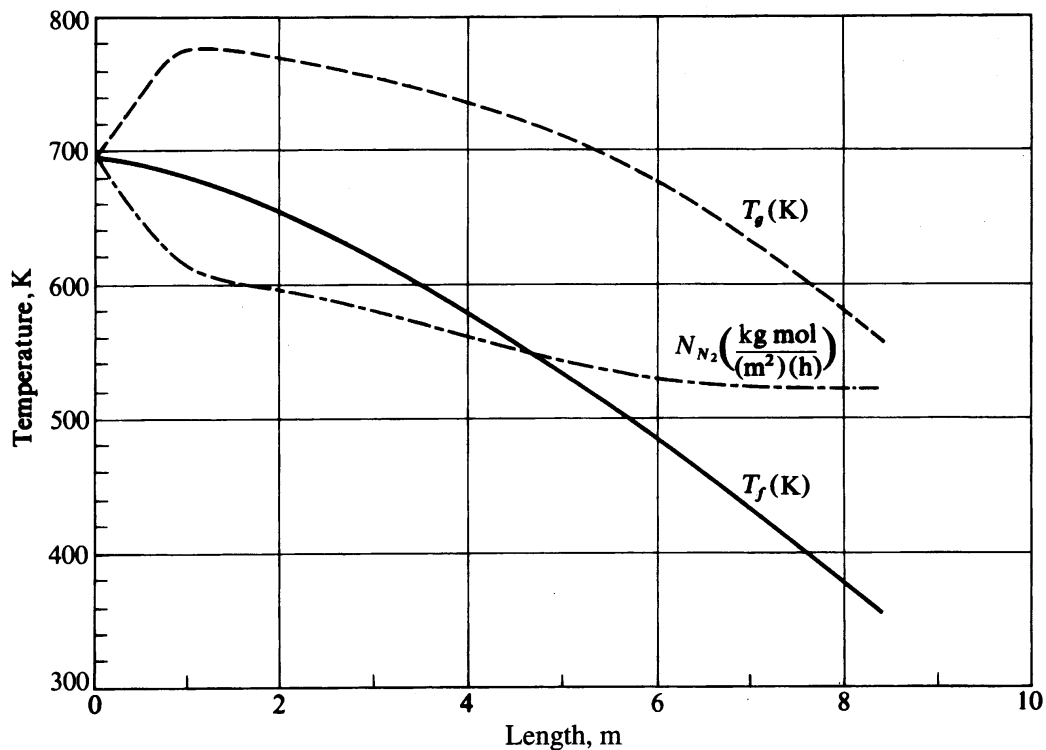


FIGURE E14.2b

Temperature and concentration profiles of the NH_3 synthesis reactor.

TABLE E14.2
Notation and data

Independent and dependent variables	
x	Reactor length, m
N_{N_2}	Mole flow rate of N_2 per area catalyst, kg mol/(m ²)(h)
T_f	Temperature of feed gas, K
T_g	Temperature of reacting gas, K
Parameters	
C_{pf}	Heat capacity of the feed gas = 0.707 kcal/(kg)(K)
C_{pg}	Heat capacity of reacting gas = 0.719 kcal/(kg)(K)
$f(\)$	Objective function, \$/year
f	Catalyst activity = 1.0
ΔH	Heat of reaction = -26,000 kcal/kg mol N_2
N	Mass flow of component designated by subscript through catalyst zone, kg mol/(m ²)(h)
N_1	Hours of operation per year = 8330
p	Partial pressure of component designated by subscript, psi; reactor pressure is 286 psia
R	Ideal gas constant, 1.987 kcal/(kg mol)(K)
S_1	Surface area of catalyst tubes per unit length of reactor = 10 m
S_2	Cross-sectional area of catalyst zone = 0.78 m ²
T_0	Reference temperature = 421°C (694 K)
U	Overall heat transfer coefficient = 500 kcal/(h)(m ²)(K)
W	Total mass transfer flow rate = 26,400 kg/h

Objective function. The objective function for the reactor optimization is based on the difference between the value of the product gas (heating value and ammonia value) and the value of the feed gas (as a source of heat only) less the amortization of reactor capital costs. Other operating costs are omitted. As shown in Murase et al., the final consolidation of the objective function terms (corrected here) is

$$f(x, N_{N_2}, T_f, T_g) = 11.9877 \times 10^6 - 1.710 \times 10^4 N_{N_2} + 704.04 T_g - 699.3 T_f - [3.4566 \times 10^7 + 2.101 \times 10^9 x]^{1/2} \quad (a)$$

Equality constraints. Only 1 degree of freedom exists in the problem because there are three constraints; x is designated to be the independent variable.

Energy Balance, Feed Gas

$$\frac{dT_f}{dx} = - \frac{US_1}{WC_{pf}} (T_g - T_f) \quad (b)$$

Energy Balance, Reacting Gas

$$\frac{dT_g}{dx} = \frac{US_1}{WC_{pg}} (T_g - T_f) + \frac{(-\Delta H)S_2}{WC_{pg}} (f) \left[K_1 \frac{(1.5)p_{N_2} p_{H_2}}{p_{NH_3}} - K_2 \frac{P_{NH_3}}{(1.5)p_{H_2}} \right] \quad (c)$$

where $K_1 = 1.78954 \times 10^4 \exp(-20,800/RT_g)$
 $K_2 = 2.5714 \times 10^{16} \exp(-47,400/RT_g)$

Mass Balance, N_2

$$\frac{dN_{N_2}}{dx} = -f \left[K_1 \frac{(1.5)p_{N_2}p_{H_2}}{p_{NH_3}} - K_2 \frac{p_{NH_3}}{(1.5)p_{H_2}} \right] \quad (d)$$

The boundary conditions are

$$T_f(x = L) = 421^\circ\text{C} \text{ (694 K)} \quad (e)$$

$$T_g(x = 0) = 421^\circ\text{C} \text{ (694 K)} \quad (f)$$

$$N_{N_2}(x = 0) = 701.2 \text{ kg mol/(h)(m}^2\text{)} \quad (g)$$

For the reaction, in terms of N_{N_2} , the partial pressures are

$$p_{N_2} = 286 \left[\frac{N_{N_2}}{1 - 2(N_{N_2}^0 - N_{N_2})} \right]$$

$$p_{N_2} = 286 \left[\frac{3N_{N_2}}{1 - 2(N_{N_2}^0 - N_{N_2})} \right]$$

$$p_{NH_3} = 286 \left[\frac{2(N_{N_2}^0 - N_{N_2})}{1 - 2(N_{N_2}^0 - N_{N_2})} \right]$$

Inequality constraints.

$$0 \leq N_{N_2} \leq 3220$$

$$400 \leq T_f \leq 800$$

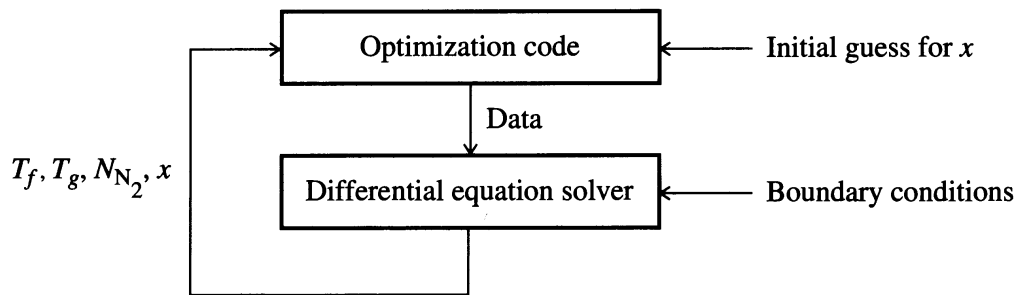
$$x \geq 0$$

Feed gas composition (mole %).

$$N_2: 21.75; \quad H_2: 62.25; \quad NH_3: 5; \quad CH_4: 4; \quad Ar: 4$$

Solution procedure. Because the differential equations must be solved numerically, a two-stage flow of information is needed in the computer program used to solve the problem. Examine Figure E14.2c. The code GRG2 (refer to Chapter 8) was coupled with the differential equation solver LSODE, resulting in the following exit conditions:

	Initial guesses	Optimal solution
N_{N_2}	646 kg mol/(m ²)(h)	625 kg mol/(m ²)(h)
Mole fraction N_2	20.06%	19.4%
T_g	710 K	563 K
T_f	650 K	478 K
x	10.0 m	2.58 m
$f(x)$	8.451×10^5 \$/year	1.288×10^6 \$/year

**FIGURE E14.2c**

Flow diagram for solution procedure, Example 14.2.

In all, 10 one-dimensional searches were carried out, and 54 objective function calls and 111 gradient calls (numerical differences were used) were made by the code.

EXAMPLE 14.3 SOLUTION OF AN ALKYLATION PROCESS BY SEQUENTIAL QUADRATIC PROGRAMMING

A long-standing problem (Sauer et al., 1964) is to determine the optimal operating conditions for the simplified alkylation process shown in Figure E14.3. Sauer and colleagues solved this problem using a form of successive linear programming. We first formulate the problem and then solve it by sequential quadratic programming. The notation to be used is listed in Table E14.3A which includes the units, upper and lower bounds, and the starting values for each x_i (a nonfeasible point). All the bounds represent economic, physical, or performance constraints.

The objective function was defined in terms of alkylate product, or output value minus feed and recycle costs; operating costs were not reflected in the function. The total profit per day, to be maximized, is

$$f(x) = C_1 x_4 x_7 - C_2 x_1 - C_3 x_2 - C_4 x_3 - C_5 x_5 \quad (a)$$

where C_1 = alkylate product value (\$0.063/octane-barrel)

C_2 = olefin feed cost (\$5.04/barrel)

C_3 = isobutane recycle costs (\$0.035/barrel)

C_4 = acid addition cost (\$10.00/per thousand pounds)

C_5 = isobutane makeup cost (\$3.36/barrel)

To form the process model, regression analysis was carried out. The alkylate yield x_4 was a function of the olefin feed x_1 and the external isobutane-to-olefin ratio x_8 . The relationship determined by nonlinear regression holding the reactor temperatures between 80–90°F and the reactor acid strength by weight percent at 85–93 was

$$x_4 = x_1(1.12 + 0.13167x_8 - 0.00667x_8^2) \quad (b)$$

The isobutane makeup x_5 was determined by a volumetric reactor balance. The alkylate yield x_4 equals the olefin feed x_1 plus the isobutane makeup x_5 less shrinkage. The volumetric shrinkage can be expressed as 0.22 volume per volume of alkylate yield so that

$$x_4 = x_1 + x_5 - 0.22x_4$$

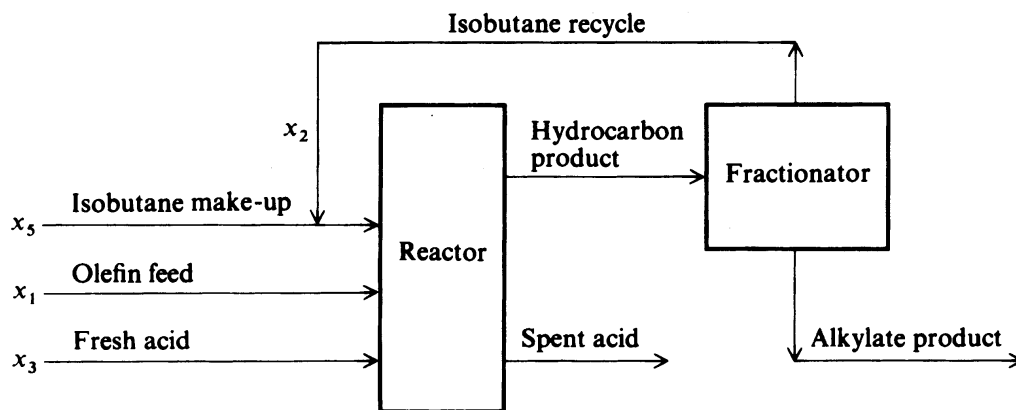


FIGURE E14.3
Alkylation flowsheet.

TABLE E14.3A

Symbol	Variable	Lower * bound	Upper bound	Starting value
x_1	Olefin feed (barrels per day)	0	2,000	1,745
x_2	Isobutane recycle (barrels per day)	0	16,000	12,000
x_3	Acid addition rate (thousands of pounds per day)	0	120	110
x_4	Alkylate yield (barrels per day)	0	5,000	3,048
x_5	Isobutane makeup (barrels per day)	0	2,000	1,974
x_6	Acid strength (weight percent)	85	93	89.2
x_7	Motor octane number	90	95	92.8
x_8	External isobutane-to-olefin ratio	3	12	8
x_9	Acid dilution factor	0.01	4	3.6
x_{10}	F-4 performance number	145	162	145

*Instead of 0, 10^{-6} was used.

or

$$x_5 = 1.22x_4 - x_1 \quad (c)$$

The acid strength by weight percent x_6 could be derived from an equation that expressed the acid addition rate x_3 as a function of the alkylate yield x_4 , the acid dilution factor x_9 , and the acid strength by weight percent x_6 (the addition acid was assumed to have acid strength of 98%)

$$1000x_3 = \frac{x_4x_9x_6}{98 - x_6}$$

or

$$x_6 = \frac{98,000x_3}{x_4x_9 + 1000x_3} \quad (d)$$

The motor octane number x_7 was a function of the external isobutane-to-olefin ratio x_8 and the acid strength by weight percent x_6 (for the same reactor temperatures and acid strengths as for the alkylate yield x_4)

$$x_7 = 86.35 + 1.098x_8 - 0.038x_8^2 + 0.325(x_6 - 89) \quad (e)$$

The external isobutane-to-olefin ratio x_8 was equal to the sum of the isobutane recycle x_2 and the isobutane makeup x_5 divided by the olefin feed x_1

$$x_8 = \frac{x_2 + x_5}{x_1} \quad (f)$$

The acid dilution factor x_9 could be expressed as a linear function of the F-4 performance number x_{10}

$$x_9 = 35.82 - 0.222x_{10} \quad (g)$$

The last dependent variable is the F-4 performance number x_{10} , which was expressed as a linear function of the motor octane number x_7

$$x_{10} = -133 + 3x_7 \quad (h)$$

The preceding relationships give the dependent variables in terms of the independent variables and the other dependent variables.

Equations (c), (d), and (f) were used as equality constraints. The other relations were modified to form two inequality constraints each, so as to take account of the uncertainty that existed in their formulation. The d_l and d_u values listed in Table E14.3B allow for deviations from the expected values of the associated variables.

Thus, the model has eight inequality constraints in addition to the three equality constraints and the upper and lower bounds on all of the variables.

$$[x_1(1.12 + 0.13167x_8 - 0.00667x_8^2)] - d_{4,l}x_4 \geq 0 \quad (i)$$

$$-[x_1(1.12 + 0.13167x_8 - 0.00667x_8^2)] + d_{4,u}x_4 \geq 0 \quad (j)$$

$$[86.35 + 1.098x_8 - 0.038x_8^2 + 0.325(x_6 - 89)] - d_{7,l}x_7 \geq 0 \quad (k)$$

$$-[86.35 + 1.098x_8 - 0.038x_8^2 + 0.325(x_6 - 89)] + d_{7,u}x_7 \geq 0 \quad (l)$$

$$[35.82 - 0.222x_{10}] - d_{9,l}x_9 \geq 0 \quad (m)$$

$$-[35.82 - 0.222x_{10}] + d_{9,u}x_9 \geq 0 \quad (n)$$

$$[-133 + 3x_7] - d_{10,l}x_{10} \geq 0 \quad (o)$$

$$-[-133 + 3x_7] + d_{10,u}x_{10} \geq 0 \quad (p)$$

To solve the alkylation process problem, the code NPSOL, a successive quadratic programming code in MATLAB, was employed.

The values of the objective function found were

$$f(x^0) = 872.3 \text{ initial guess}$$

$$f(x^*) = 1768.75$$

TABLE E14.3B

Deviation parameter	Value
$d_{4,i}$	99/100
$d_{4,u}$	100/99
$d_{7,i}$	99/100
$d_{7,u}$	100/99
$d_{9,i}$	9/10
$d_{9,u}$	10/9
$d_{10,i}$	99/100
$d_{10,u}$	100/99

TABLE 14.3C

Variable	Optimal value	Variable	Optimal value
x_1	1698.1	x_6	90.115
x_2	15819	x_7	95.000*
x_3	54.107	x_8	10.493
x_4	3031.2	x_9	1.5618
x_5	2000.0*	x_{10}	153.54

*At bound.

TABLE E14.3D

Constraint	Value at x^*	Constraint	Value at x^*
1(i)	0.33	7(o)	60.9
2(j)	0.18×10^{-11}	8(p)	1.91
3(k)	-0.22×10^{-12}	9(c)	0.29×10^{-10}
4(l)	573	10(d)	0.45×10^{-12}
5(m)	0	11(f)	-0.57×10^{-13}
6(n)	0.45×10^{-12}		

Tables E14.3C and E14.3D list values of the variables at x^* (rounded to five significant figures) and the constraints, respectively, at the optimal solution.

Note that the value of the isobutane makeup x_5 is at its upper bound.

EXAMPLE 14.4 PREDICTING PROTEIN FOLDING

Although the field of molecular modeling is relatively new, it is expanding rapidly with advances in computational power. The appeal of molecular modeling lies in the wealth of potential theoretical developments and practical applications in drug design, food chemistry, genome analysis, and biomedical engineering. A particularly challenging problem involves the prediction of protein folding that can be treated as a global and combinatorial optimization problem. Proteins are three-dimensional structures whose configuration in principal can be predicted from information about a particular amino

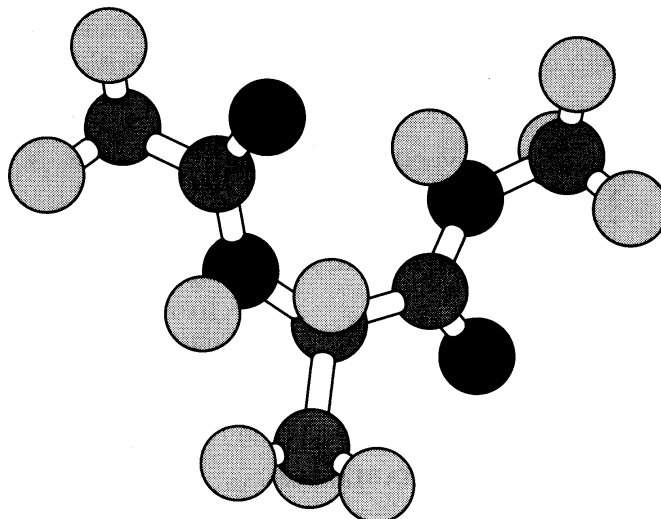


FIGURE E14.4a
Conformation of *N*-acetyl-*N'*-methyl-alanineamide
(Courtesy of C. A. Floudas).

acid sequence along with the environmental conditions. Naturally occurring proteins are composed of 20 different amino acid compounds with different side chains and a backbone of repeating units connected by peptide bonds. Covalent bond angles and interatomic forces cause the chain to form and twist in a unique way in three dimensions for each protein. Figure E14.4a illustrates a computer-generated model of the protein *N*-acetyl-*N'*-methyl-alanineamide.

Once the folded sequence is known, the biological and chemical properties of the protein can be predicted. In the development of drugs, for example, the intended target in the human body is a particular protein of known structure whose behavior can be altered (for the better) when a drug molecule binds to a receptor site on the target molecule.

In spite of the complexity of the protein-folding problem, prediction of folding rests on a simple thermodynamic concept: The folded configuration can be identified by minimizing the global free energy of the molecule. Two main components exist, namely the unsolvated potential energy and the solvation energy, the sum of which must be minimized.

$$E \equiv E_{\text{Total}} = E_{\text{Unsol}} + E_{\text{Sol}} \quad (a)$$

This criterion requires a search through a nonconvex multidimensional conformation space that contains an immense number of minima. Optimization techniques that have been applied to the problem include Monte Carlo methods, simulated annealing, genetic methods, and stochastic search, among others. For reviews of the application of various optimization methods refer to Pardalos et al. (1996), Vasquez et al. (1994), or Schlick et al. (1999).

The example considered here involves the use of a branch-and-bound global optimization algorithm known as α BB (Adjiman et al., 1998) as carried out by Klepeis et al. (1998) who calculated the minimum energy for a number of peptides. To simplify an inherently very complicated optimization problem, particularly in view of the limited data known about solvation parameters, they formulated the energy minimization

problem using the dihedral angles (assuming the covalent bond lengths and bond angles fixed at their equilibrium values) as the optimization variables as follows:

$$\text{Minimize: } E(\phi_i, \psi_i, \omega_i, \chi_i^k, \theta_j^N, \theta_j^C) \quad (b)$$

Subject to:

$$-\pi \leq \phi_i \leq \pi, \quad i = 1, \dots, N_{\text{Res}}$$

$$-\pi \leq \psi_i \leq \pi, \quad i = 1, \dots, N_{\text{Res}}$$

$$-\pi \leq \omega_i \leq \pi, \quad i = 1, \dots, N_{\text{Res}}$$

$$-\pi \leq \chi_i^k \leq \pi, \quad i = 1, \dots, N_{\text{Res}}$$

$$k = 1, \dots, K^i$$

$$-\pi \leq \phi_j^N \leq \pi, \quad i = 1, \dots, J_N$$

$$-\pi \leq \phi_j^C \leq \pi, \quad j = 1, \dots, J_C$$

where E represents the total of the potential energy function and the free energy of solvation. E_{Unsol} is

$$\begin{aligned} E_{\text{Unsol}} &= \sum_{(ij) \in \text{ES}} \frac{q_i q_j}{r_{ij}} && \text{(Electrostatic contribution-ES)} \\ &+ \sum_{(ij) \in \text{NB}} F_{ij} \frac{A_{ij}}{r_{ij}^{12}} - \frac{C_{ij}}{r_{ij}^6} && \text{(Nonbonded contribution-NB)} \\ &+ \sum_{(ij) \in \text{HB}} \frac{A'_{ij}}{r_{ij}^{12}} - \frac{B_{ij}}{r_{ij}^{10}} && \text{(Hydrogen bonded contribution-HB)} \\ &+ \sum_{k \in \text{TOR}} \left(\frac{E_{o,k}}{2} \right) (1 + \cos n_k \theta_k) && \text{(Torsional contribution-TOR)} \\ &+ \sum_{l \in \text{CL}} 100 \sum_{i=1}^{i=3} (r_{il} - r_{io})^2 && \text{(Cystine loop-closing contribution-CL)} \\ &+ \sum_{l \in \text{CT}} \left(\frac{E_{o,l}}{2} \right) (1 - \cos n_l \chi_l) && \text{(Cystine torsional contribution-CT)} \\ &+ \sum_{p \in \text{PRO}} E_p && \text{(Proline internal contribution-PRO)} \end{aligned}$$

where: A_{ij} = nonbonded parameter specific to the atomic pair
 A'_{ij} = hydrogen-bonded parameter specific to the atomic pair
 B_{ij} = hydrogen-bonded parameter specific to the atomic pair
 C_{ij} = nonbonded parameter specific to the atomic pair
 $E_{o,k}$ = parameter corresponding to torsional barrier energy for a dihedral angle θ_k
 $E_{o,l}$ = parameter corresponding to torsional barrier energy for a dihedral angle χ_l involved in cystine loop closing

- E_p = fixed internal energy for each proline residue in the protein
 F_{ij} = coefficient equal to 0.5 for one to four interactions and equal to 1.0 for one to five and higher interactions
 i = index denoting the sequence of amino acid residues in the peptide chain
 j = index denoting the dihedral angles of the amino acid end group
 J_C = number of carboxylic end groups
 J_N = number of dihedral angles of the end group
 k = index denoting the dihedral angles of the side chains for the i th amino acid residue
 K^i = number of angles on the side chains
 N_{Res} = number of amino acid residues
 n_k = symmetry type for θ_k
 n_l = symmetry type for χ_l
 q_i = dipole parameter for atom i
 q_j = dipole parameter for atom j
 r_{ij} = the interatomic distance in the atomic pair ij
 r_{il} = actual interatomic distance
 r_{io} = required interatomic distance
 $\theta_i, \psi_i, \omega_i$ = dihedral angles along the backbone of the peptide chain
 χ_i^k = side chain dihedral angle
 θ_j^C = dihedral angles of the carboxy end groups
 θ_j^N = dihedral angles of the amino end groups

To reduce undesirable perturbations in the minimization, and for other reasons explained in Klepeis et al., the first term on the right-hand side of Equation (a) was minimized before adding the contribution from the second term. The specific details and parameters of problem (b) can be found in Klepeis et al.

Klepeis et al. extended the α BB optimization algorithm to guarantee convergence to the global optimum of a nonlinear problem with twice differentiable functions. Without such a guarantee, the outcome depends too heavily on the allocated initial conditions for the molecular configuration. The α BB optimization algorithm brackets the global minimum solution by developing converging lower and upper bounds. These bounds are refined by successively partitioning the region for search. Upper bounds on the global minimum are obtained by local minimizations of the original energy function E . Lower bounds L are obtained by minimizing convex lower-bounding functions that are constructed by adding to E the sum of separable quadratic terms such as

$$\sum_{i=1}^{N_{\text{Res}}} \alpha_{\psi, i} (\psi_i^L - \psi_i) (\psi_i^U - \psi_i)$$

for each angle (6 terms are added).

The α represent nonnegative parameters that must be greater than or equal to the negative one-half of the minimum eigenvalue of the Hessian of E over the defined domain. These parameters can be estimated by the solution of an optimization problem or by using the concept of the measure of a matrix (Maranas and Floudas, 1994). The net result is to make L convex. A useful property of L is that the maximum separation between L and E is bounded and is proportional to α and to the square of the diagonal of the successive box constraints, so that convergence to a global optimum occurs.

At any stage, once solutions for the upper and lower bounds have been established, the next step is to modify the bounding problems for the next iteration. This is accomplished by successively partitioning the initial domain into smaller subdomains. The default partitioning strategy used in the algorithm involves successive subdivision of the original hyper-rectangle by halving on the midpoint of the longest side (bisection). A nonincreasing sequence for the upper bound is found by solving the nonconvex problem E locally and selecting it to be the minimum over all the previously recorded upper bounds.

Initially Klepeis et al. allowed the dihedral angles to vary over the entire $[-\pi, \pi]$ domain. It was found, however, that the problem required intensive computational effort (Androulakis et al., 1997). A reduction of the domain space was therefore proposed by setting limits based on the actual distributions of the dihedral angles. Obviously, for the algorithm to be successful, these reductions could not exclude the region of the global minimum conformation.

The computational requirement of the α BB algorithm depends on the number of variables on which branching occurs. The most important variables are those variables that substantially influence the nonconvexity of the surface and the location of the global minimum. In the protein-folding problem, the backbone dihedral angles (ϕ and ψ) are the most influential variables. Therefore, in very large problems, to further reduce the dimensions of the problem, only these variables were involved in the optimization.

Figure E14.4b shows the results of the application of the optimization strategy to solvated N -acetyl- N' -methyl-alanineamide. Level sets of the deviations of the total energy from the global minimum are shown as solid and dashed lines at 1, 2, 5, and

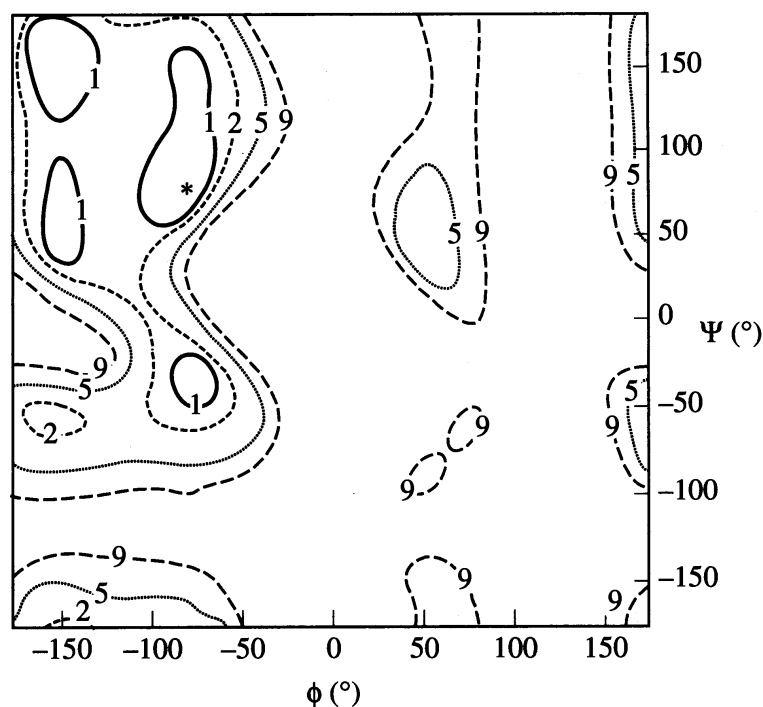


FIGURE E14.4b

Surface of the objective function obtained in determining the structure of N -acetyl- N' -methyl-alanineamide; * is the minimum, and the level sets denote deviations from the minimum (in kcal/mol).

9 kcal/mol, respectively; *designates the global minimum. Klepeis et al. list the various components of the total energy as a function of the amino residues for the protein. Only qualitative comparisons can be made with actual proteins because of the lack of experimental data.

EXAMPLE 14.5 OPTIMIZATION OF LOW-PRESSURE CHEMICAL VAPOR DEPOSITION REACTOR FOR THE DEPOSITION OF THIN FILMS

The manufacture of microelectronic devices involves the sequencing of processes involving thin film deposition, patterning, and doping, only the first of which is discussed here. The formation of the films is performed by a variety of techniques, including physical and chemical processes. One of the most versatile of these methods is chemical vapor deposition (CVD), which involves reacting gases flowing over a wafer to form the desired film. Energy for the reaction is provided by heat or from a plasma. CVD requires the diffusion of gaseous reactants to the hot substrate (wafer), adsorption, reaction, desorption, and diffusion of the gaseous products back into the bulk gas. The net result of the process is formation of a film on the substrate. One common configuration used for CVD stacks the wafers in a tube such as that shown in Figure E14.5a, with heating provided by furnace elements (Middleman and Hochberg, 1993). The low-pressure chemical vapor deposition (LPCVD) reactor allows a large number of wafers to be processed in one batch, yielding good film thickness and composition uniformity.

The LPCVD reactor shown in Figure E14.5a operates at pressures of 0.1–1 torr. The close stacking of the wafers allows for a large throughput while taking advantage of the fact that at these low pressures gas diffusivities are high. This arrangement allows good transport of gases into the region between the wafers (the interwafer region) and hence good radial uniformity of deposition. The flow in the region between the wafer edges and the reactor wall (the annular region) is laminar at typical LPCVD conditions. The reactor walls as well as the wafers are hot so that radial temperature gradients are small. The nonuniformity of growth rates in the radial direction is thus minimized.

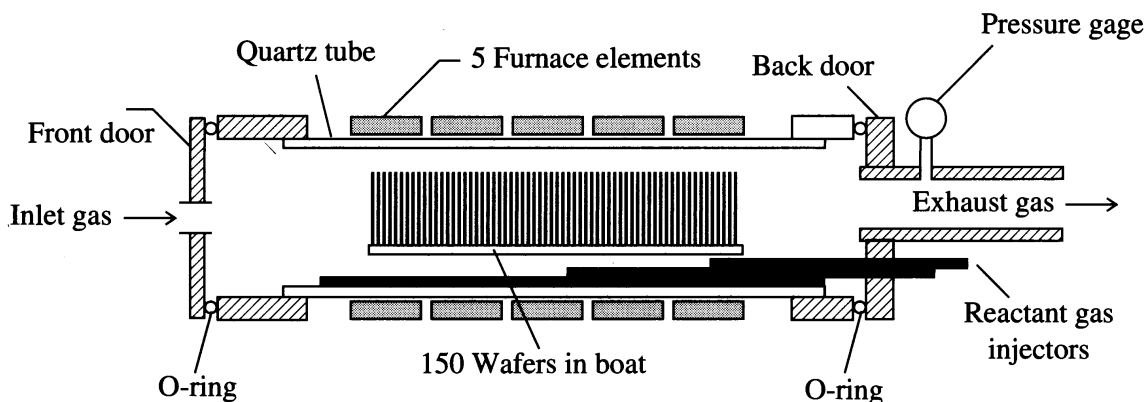
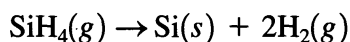


FIGURE E14.5a

A typical multiwafer hot-wall low-pressure chemical vapor deposition reactor.

In most microelectronics fabrication factories (“fabs”), LPCVD of polycrystalline silicon (poly-Si) is carried out by the decomposition silane



The gas-solid reaction rate is modeled by the nonlinear expression

$$R = \frac{k_1 p_{\text{SiH}_4}}{1 + k_2 p_{\text{H}_2}^{1/2} + k_3 p_{\text{SiH}_4}} \quad (a)$$

where R = the reaction rate
 p = the partial pressure
 k_1, k_2, k_3 = rate constants

The rate expression is based on adsorption–desorption equilibrium at the substrate surface with an additional term ($k_2 p_{\text{H}_2}$) representing H_2 gas inhibition. The rate constants can be estimated by regression of R with the two partial pressures using experimental data (Roenigk and Jensen, 1985).

Model equations. Fundamental process models are very useful in optimizing the design and operation of LPCVD systems. A fundamental model of an LPCVD reactor similar to Figure E14.5a was presented by Jensen and Graves (1983) and included the following simplifying assumptions:

1. The reactor shown in Figure E14.5a, has no radial temperature gradients because its walls and substrate are heated and slow reaction rates imply small heats of reaction.
2. The axial temperature gradient is fixed by the furnace settings because the gas heat-up lengths are small and most heat transfer occurs by radiation at LPCVD conditions.
3. There is no axial variation of gas-phase composition in the interwafer region between any two consecutive wafers because the interwafer spacing is small.
4. There is no radial variation of gas-phase composition in the annular region because the annular region is small, and because there is rapid diffusion at LPCVD conditions.
5. The gas phase is in steady state, because CVD growth processes are slow compared to gas phase dynamics.

These five assumptions were used by Jensen and Graves (1983) and were also employed in the design study of Setalvad and colleagues (1989). Define N_r and N_z as the molar respective fluxes of silane in the r and z directions, Δ as the interwafer spacing, and x_1 as the mole fraction of silane in the gas phase. The mass transport equations in the r and z directions that describe the diffusion of silane consist of two coupled partial differential equations. In Setalvad and colleagues (1989), the partial differential equations in the r and z directions were converted to ordinary differential equations by assuming the axial transport (N_z) only occurred in the annular region, whereas the radial transport (N_r) only occurred in the interwafer region (see Figure E14.5b). The LPCVD model thus is as follows.

Interwafer Region

$$\frac{\Delta}{r} \frac{d}{dr}(rN_{r1}) = -2R \quad (b)$$

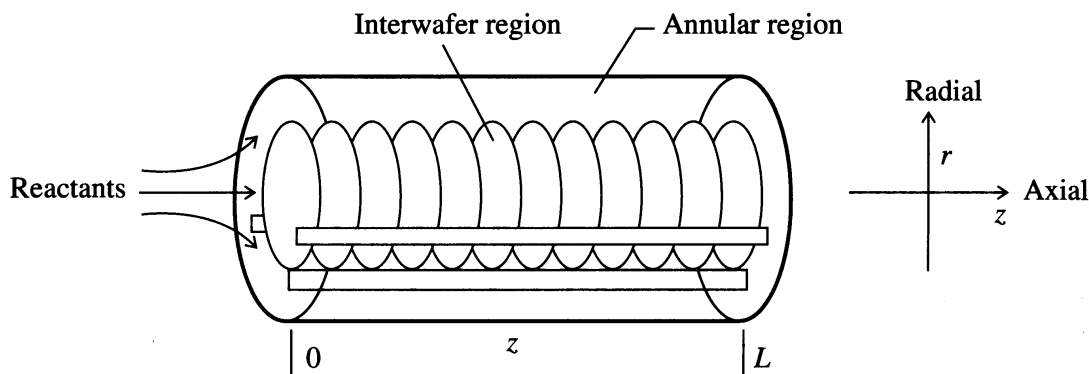


FIGURE E14.5b
LPCVD reactor geometry with interwafer and annular regions.

with boundary condition:

$$\left. \frac{dx_1}{dr} \right|_{r=0} = 0 \quad \text{and} \quad x_1(r_w^-) = x_1(r_w^+) \quad (c)$$

where r_w = the wafer radius, and

+, - refers to an infinitesimal distance in positive/negative r direction

Annular Region

$$\frac{dN_{z_1}}{dz} = - \frac{2R}{(r_i^2 - r_w^2)} \left[r_i(1 + a) + \frac{r_w^2}{\Delta} \eta \right] \quad (d)$$

where r_i = the tube radius

a = the area of the wafer holder plus wafers relative to the reactor tube area

The fluxes are related to the mole fraction through Fick's law (c is total concentration of gas, and D is the diffusivity of the silane in the gas phase):

$$N_{r_1} = cD \frac{dx_1}{dr} \quad (e)$$

$$N_{z_1} = cD \frac{dx_1}{dz} \quad (f)$$

Boundary conditions for the annular region are given at the inlet ($z = 0$) and the tube exit ($z = L$):

$$N_{z_1} \Big|_{z=0} = v_0 c_0 x_{10} \quad \text{and} \quad \left. \frac{dx_1}{dz} \right|_{z=L} = 0 \quad (g)$$

where $v_0 c_0 x_{10}$ represents the product of gas velocity, total concentration, and mole fraction silane at the inlet.

The effectiveness factor η is defined as

$$\eta = \frac{2 \int_0^{r_w} rR(r)dr}{r_w^2 R|_{r_w}} \quad (h)$$

η is the ratio of the average rate of deposition on a wafer to that at its edge, so it is a measure of the uniformity of deposition. The rate R at r_w varies in the z direction, hence η is a function of axial distance z . The effectiveness factor represents the radial uniformity of deposition. When surface reaction is rate-controlling, $\eta = 1$, and when $\eta < 1$, diffusion resistance comes into play.

Optimization of the reactor. The nonlinear ordinary differential equations and boundary conditions in the model can be put in dimensionless form and converted to algebraic equations using orthogonal collocation (Finlayson, 1980). Setalvad and coworkers (1989) used these algebraic equations as constraints in formulating a nonlinear programming problem to study the effects of temperature, flow parameters, reactor geometry, and wafer size on the LPCVD process, particularly the uniformity of silicon deposition. Strategies were devised to determine the potential improvements in the system performance by using optimum temperature staging and reactant injection schemes. Figure E14.5c shows the inputs and performance measures for the reactor that can be optimized to maximize the film growth rate (production rate), subject to constraints on radial film uniformity (on each wafer), as well as axial uniformity (wafer-to-wafer).

The growth rate is quite sensitive to the axial temperature profile. An axial temperature profile that increases along the reactor because it improves the deposition uniformity is commonly used in industry. The temperature of each successive zone in the furnace (defined by the furnace elements in Figure E14.5a) can be adjusted by voltage applied to variac heaters. The zone temperatures are assumed constant within each zone, $T_j, j = 1, \dots, n_{tz}$, where n_{tz} is the number of temperature zones to be used,

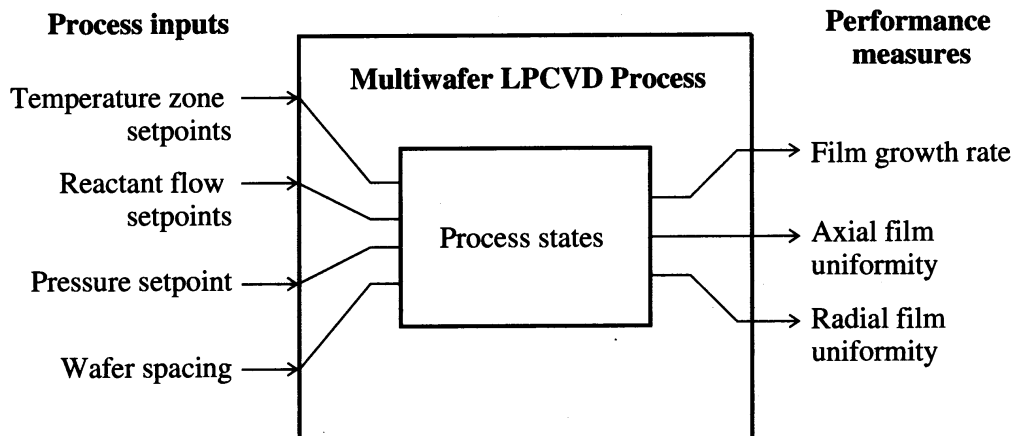


FIGURE E14.5c
Multiwafer LPCVD reactor process inputs and outputs.

typically three to five. The optimization procedure was initiated with all T_j values equal to 880 K. The objective function to be maximized in this case was

$$f(\mathbf{T}) = \int_0^L [G(\mathbf{T}, z)]^2 dz \cong \sum_{i=1}^N [G_i(\mathbf{T}, z)]^2 \Delta z_i \quad (i)$$

G_i is determined by averaging the growth rate given by Equation (a) over the wafer surface, which is then integrated over the axial direction to compute $f(\mathbf{T})$. G is measured in $\text{\AA}/\text{min}$, \mathbf{T} is the set of temperatures, and z_i ($i = 1, N$) are locations along the reactor at which the model is solved to obtain the rates G_i . $f(\mathbf{T})$ is a uniformly weighted sum of the deposition rates over the entire reactor ($N =$ number of increments) obtained via the LPCVD model, thus representing the throughput of the reactor.

The objective function is maximized subject to the following inequality constraints:

1. The maximum allowable axial variation in growth rate is 5% of the maximum rate,

$$V = \frac{\max(G_i) - \min(G_i)}{\max(G_i)} \leq 0.05 \quad (j)$$

2. At no point should the radial variation in growth rate be greater than 5%; in terms of effectiveness factors,

$$\eta_i \geq 0.95, \quad i = 1, \dots, n_{rz} \quad (k)$$

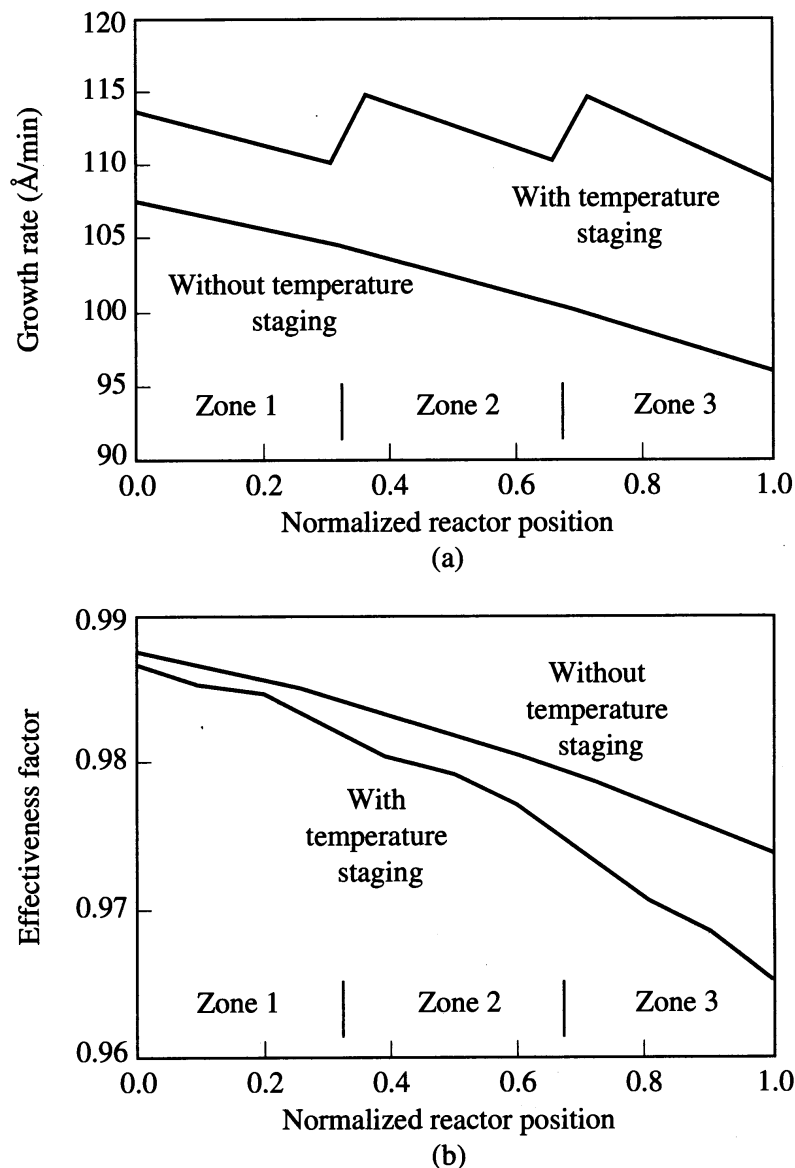
3. The temperature in each zone is restricted to

$$880 \text{ K} \leq T_i \leq 890 \text{ K}, \quad i = 1, \dots, n_{rz} \quad (l)$$

This last constraint is imposed so that the grain size and other temperature-dependent material properties of the grown film and also its step coverage do not show excessive variations.

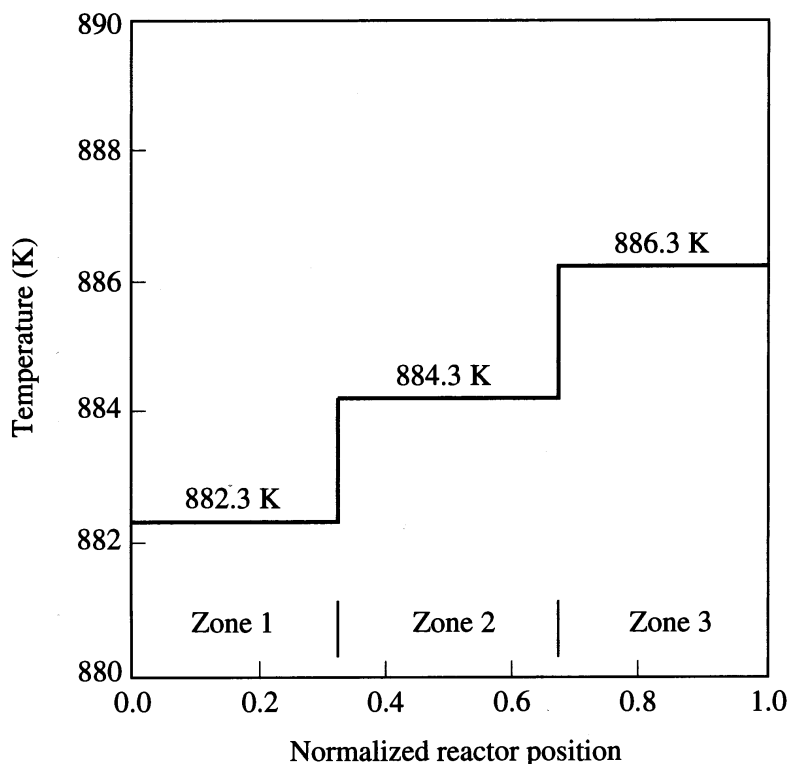
The nonlinear programming problem based on objective function (i), model equations (b)–(g), and inequality constraints (j)–(l) was solved using the generalized reduced gradient method presented in Chapter 8. See Setalvad and coworkers (1989) for details on the parameter values used in the optimization calculations, the results of which are presented here.

In Figure E14.5d the performance of the reactor with operation with each of three temperature zones at their optimal values can be compared with the isothermal case ($T_i = 880 \text{ K}$). The optimization routine increased the temperature of zone 3 the most, followed by zone 2 (see Figure E14.5e). The optimization strategy increased the value of $f(\mathbf{T})$ while decreasing the maximum axial growth rate variation. The temperatures were increased from the initial value (880 K) until the axial rate variation (m_i) between the beginning and the end of zone 3 reached the 5% limit. Reactant depletion causes the sharp drop-off in rate within the zone. This effect of reactant depletion increases noticeably from zone 1 to zone 3 (Figure E14.5d). The temperature in zone 2 could be decreased so that less reactant is consumed in this zone and more is available for zone 3. However, the resulting lower rates in zone 2 cause the axial rate variation between the end of this zone and the beginning of zone 3 to exceed the 5% limit.

**FIGURE E14.5d**

Reactor performance with and without optimized temperature staging.

Optimum reactant injection. An alternative to using temperature staging is to provide a sudden increase in the partial pressure of SiH_4 , using the reactant gas injectors shown in Figure E14.5a, so that additional reactant is fed into the reactor at different points along its length. Sudden increases in growth rate at the injection points result without the disadvantage of excessive rate drop-off due to reactant depletion, as seen for the case of temperature staging. For modeling purposes the original reactor with two reactant injection ports can be considered to consist of three smaller reactors or subreactors. Predicting the performance of the reactor then involves consecutively solving the modeling equations for each of the subreactors; see Setalvad and coworkers (1989) for more details.

**FIGURE E14.5e**

The optimum temperature profile for three stages.

The optimization problem for the case of sudden injection of SiH_4 involves as independent variables the total gas flow velocities:

$$v_{0i}, \quad i = 1, \dots, n_{\text{inj}}$$

based on the total amount of gas injected, and

$$x_{1i}, \quad i = 1, \dots, n_{\text{inj}}$$

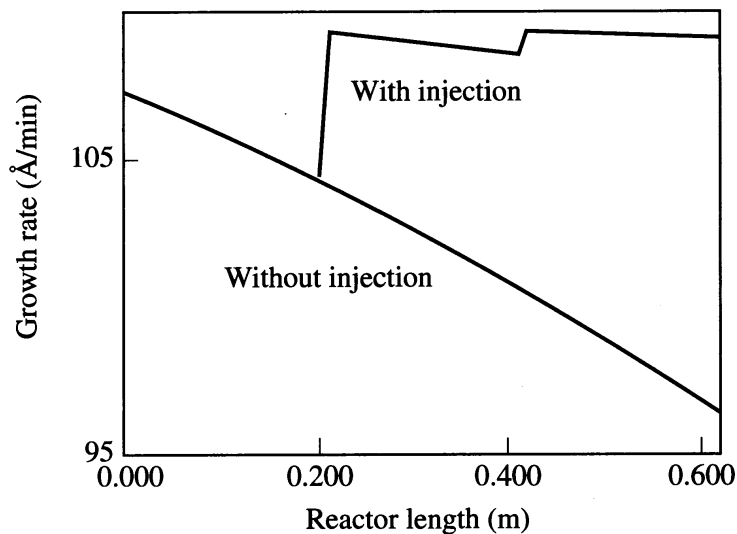
the mole fractions of the reactant silane in each injection stream. Here n_{inj} is the number of injection points. Two intermediate injection points were considered, giving four independent variables to be adjusted (two velocities and two mole fractions). This formulation was thought to be a reasonable balance between improved reactor performance and the resulting greater design complexity.

The objective function to be maximized was essentially the same as before except that the rate G was now a function of v_{0i} and x_{1i} instead of T_i , that is,

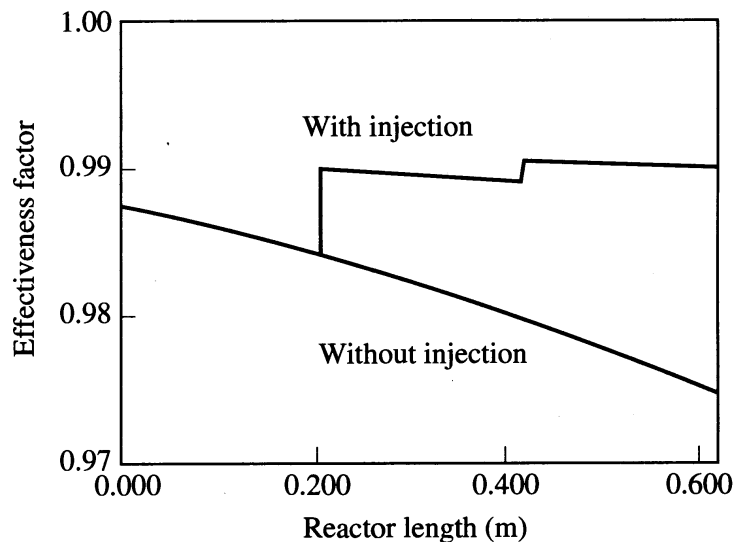
$$f(\mathbf{v}_0, \mathbf{x}_1) = \int_0^L G[(\mathbf{v}_0, \mathbf{x}_1, z)]^2 dz \approx \sum_{i=1}^N [G_i(\mathbf{v}_0, \mathbf{x}_1, z_i)]^2 \Delta z_j \quad (m)$$

The uniformity inequality constraints [Equations (j)–(l)] were again included in the problem. Additionally, the bounds on the variables were

$$0.0 \leq x_{1i} \leq 1.0, \quad i = 1, \dots, n_{\text{inj}} \quad (n)$$



(a)



(b)

FIGURE E14.5f
Reactor performance with optimum staged injection.

and

$$0.03 \leq v_{0i} \leq 0.5, \quad i = 1, \dots, n_{\text{inj}} \quad (c)$$

The optimum reactor growth rate and effectiveness factor are shown in Figure E14.5f. As expected, the optimization code adjusted v_{01} first because the deposition was more sensitive to flow velocities. After v_{01} reached its upper bound, x_{11} increased until the axial uniformity constraint was reached, that is, the difference in growth rate between the end of the first zone and the beginning of the second was equal to 5% of the inlet value (see Figure E14.5f) according to constraint (j). However, for injection point 2, the rates did not change by 5% between the injection points. Maximizing overall growth rate was more easily solved by increasing x_{12} . The effectiveness factors (Figure E14.5f), unlike those in the previous temperature profile optimization (Figure E14.5d) stayed nearly constant along the axial direction.

Setalvad and coworkers (1989) also evaluated nonuniform interwafer spacing in the reactor to improve deposition uniformity and increase the reactor throughput. Optimal interwafer spacings were smaller toward the reactant inlet to take advantage of the larger reactant concentration in this region, and larger at the end of the reactor where reactant depletion and hydrogen production inhibited the polysilicon deposition. This scheme exhibited decreased sensitivity of the process to gas flow rate variations when compared with the uniformly spaced wafer case.

A subsequent study by Badgwell and colleagues (1992) used a more detailed deposition model that was verified on industrial-scale LPCVD equipment. Badgwell and colleagues showed a sharp decrease in deposition uniformities for a wafer to reactor diameter ratio of about 0.5. This outcome suggested that it may not be wise to use existing reactors for larger wafer sizes. Furthermore, the reactor tubes that would then be necessary may have to be inordinately large and, in view of the low pressures, inordinately thick to be economical.

EXAMPLE 14.6 REACTION SYNTHESIS VIA MINLP

Process synthesis involves intelligent decision making to select a process design whose configuration and operating states are optimal in some sense. The development of mixed-integer nonlinear programming (MINLP) algorithms has greatly expanded the scope of quantitative synthesis because we can now treat synthesis problems involving both continuous and discrete variables. In this example, we demonstrate the use of MINLP in the synthesis of a hydrodealkylation (HDA) process (Douglas, 1988) as carried out by Phimister and colleagues (1999).

The fundamental decisions in the synthesis of a multistep process that involves individual reactor units connected in serial and parallel configurations as well as recycle pertain to how the units will be connected. In addition, however, we must consider (for a steady-state process)

1. What the feeds and their quantities should be.
2. What reaction paths to avoid.
3. What products should be made and in what quantity.
4. What the flow rates should be.
5. What variables affect the products.
6. How to maintain flexibility.
7. Safety issues.

We consider only factors 1 through 4 in what follows. Phimister and colleagues decomposed the strategy of reactor process design so that a mathematical statement of the synthesis problem could be formulated in terms of an objective function and constraints. At the initial stage of the decision making, the designer is presumed to have limited information about possible reaction paths as reflected in kinetic models, costs of raw materials, selling prices of products, and the desired plant production. The MINLP problem formulation in this example includes (1) binary decision variables designating whether or not a connection exists between reactors, (2) specification of continuous variables corresponding to flow rates, and (3) prespecification of the extent of conversion of a reactant.

From the topographical viewpoint illustrated in Figure E14.6a the process comprises a set of reactor–separator sections that connect a set of component feeds (specified as source nodes) to component products (specified as destination nodes). Each section is a prescribed sequence of reactors and associated separation units, and sev-

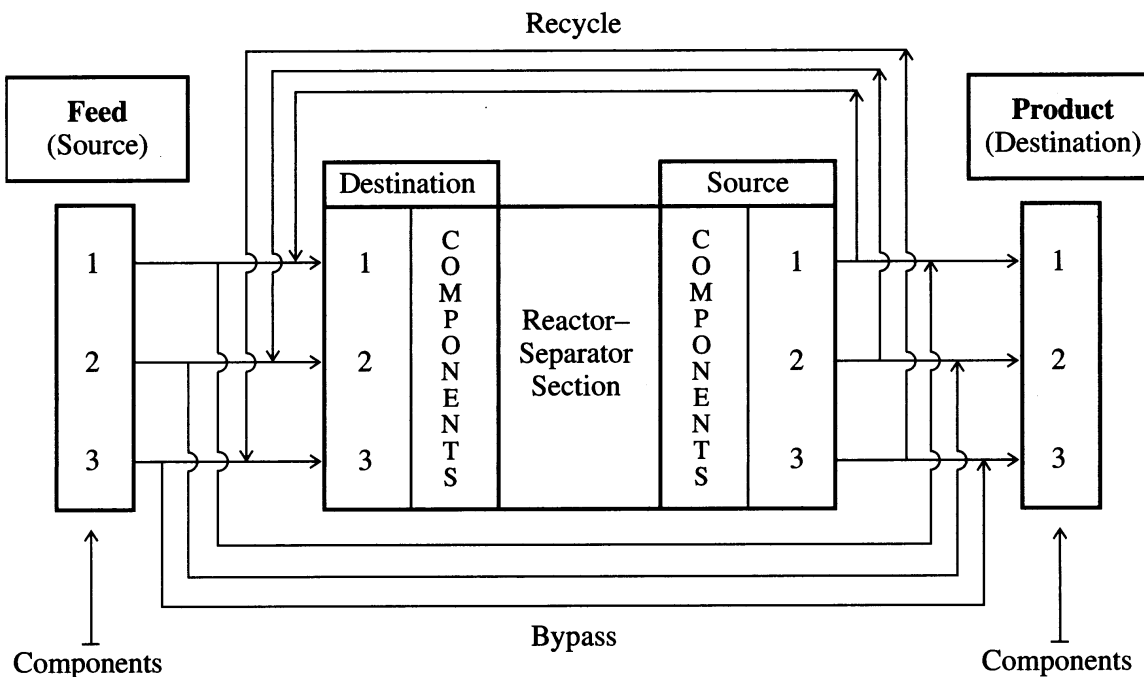


FIGURE E14.6a
Schematic of reactor-separation process.

eral sections may be interconnected, although for simplicity in presentation, we show only one such section in this example. The details of the design of the reactors and separators constituting a section are determined after the MILNP problem is solved. A source node defines the site from which a component is supplied, and a destination node defines the site at which a component is required in the process. In the initial topography (see Figure E14.6a) for the process, all of the components in the nodes are connected via directed paths, except that usually no feedback exists from a process component destination to a process component source. By use of binary variables in the constraints (as we will show later), a set of paths can be eliminated from the MINLP problem formulation, thus simplifying the topography. Various connections may be required, such as a particular feed from an external source node to a section, and various connections may be deleted, such as a bypass path from a process source to a process destination or a recycle path for a section.

The notation used in this example for the connections is as follows:

- D = the set of destination nodes $\{1, 2, \dots, e\}$ including reactor-separator sections (recycle) or flows exiting the overall process
- e = the process exit stream
- f = the process feed stream
- n = the number of components
- N = the total number of plant reactor-separator sections
- Q = the set of components, $i = 1, 2, \dots, n$
- S = the set of source nodes $\{f, 1, 2, \dots, N\}$
- X = fraction conversion of toluene
- $y_{i,j,k}$ = a binary variable (0, 1) in which the subscript i designates the chemical component, j denotes the source node, and k denotes the destination node
- ψ = selectivity of toluene converted to benzene

As examples of the notation for the binary variables, $y_{\text{CH}_4, f, 2} = 1$ means that methane in the feed stream goes to the reactor-separator section labeled No. 2, $y_{\text{H}_2, f, e} = 0$ means that hydrogen in the feed stream does not go directly to the exit stream, and $y_{\text{CO}, 1, 1}$ means that carbon monoxide is recycled in the reactor-separator section labeled No. 1.

Stream flow rates F that exit are designated with the same set of subscript indices, i, j , and k , that have the same meaning as that used for the binary variables. Negative flow rates are not allowed ($F_{i,j,k} \geq 0$). Constraints such as

$$F_{i,j,k} - Uy_{i,j,k} \leq 0$$

where U is the largest flow rate allowed between two sites, place an upper bound on a flow rate.

The reaction(s) in a reactor-separator section is accounted for by an equality constraint(s) such as for the case of $A \rightarrow B$ in section 1

$$\sum_k F_{A, 1, k} = (1 - X) \sum_j F_{A, j, 1}$$

$$\sum_k F_{B, 1, k} = \sum_j F_{B, j, 1} + X \sum_j F_{A, j, 1}, \quad j \in S, k \in D$$

where X is the fraction conversion of A to B .

In the HDA process represented in Figure E14.6.b (Douglas (1988)), the reactions are

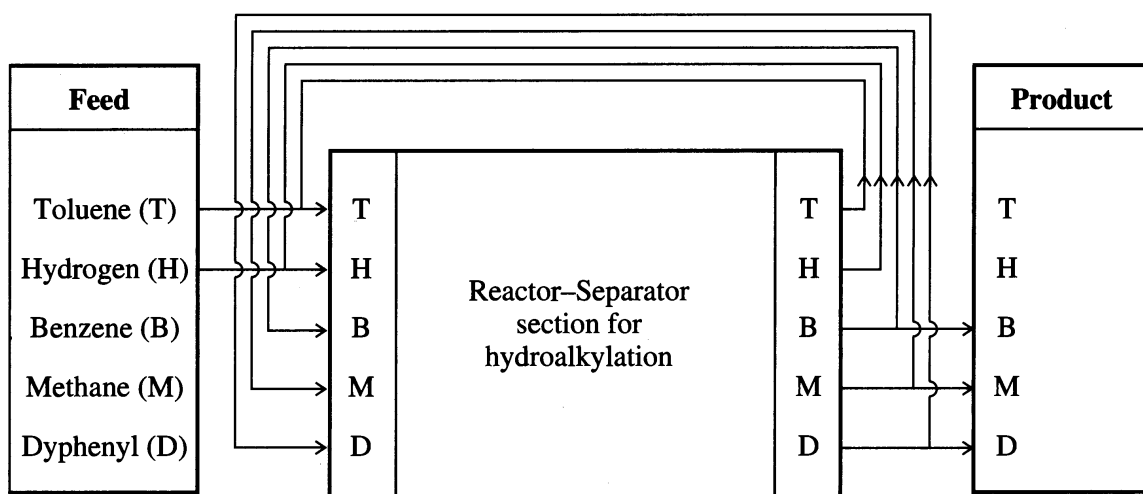
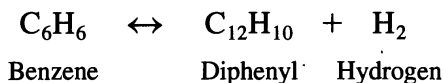
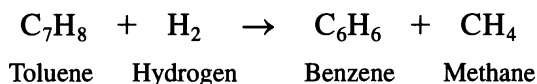


FIGURE E14.6b
Component flow diagram.

From Figure E14.6b you can see that five components exist for the one reactor–separator section, and 20 binary variables and stream flows (continuous variables) occur in the initial model. Let us summarize the model proposed by Phimister and colleagues. Other details are in Douglas.

Constraints Involving Binary Variables

Bypass prohibited:

$$\sum_i y_{i,f,e} = 0, \quad i \in Q \quad (a)$$

Only toluene and hydrogen are feeds:

$$y_{C_6H_6,f,1} + y_{C_{12}H_{10},f,1} + y_{CH_4,f,1} = 0 \quad (b)$$

$$y_{C_7H_8,f,1} = 1 \quad (c)$$

$$y_{H_2,f,1} = 1 \quad (d)$$

Only benzene, methane, and diphenyl leave the process

$$y_{C_6H_6,1,e} = 1 \quad (e)$$

$$y_{CH_4,1,e} = 1 \quad (f)$$

$$y_{C_{12}H_{10},1,e} = 1 \quad (g)$$

No toluene exits the process to a destination

$$\sum_j y_{C_7H_8,j,e} = 0 \quad (h)$$

For the reactor–separator section, all source nodes must have a destination

$$\sum_k y_{i,1,k} \geq 1, \quad k \in D, \quad \forall i \quad (i)$$

Constraints Representing the Model

Douglas (1988, Appendix B) fit the selectivity of the data ψ versus X given in the 1967 AIChE Student Contest Problem to get

$$\psi = \frac{\text{Moles benzene formed}}{\text{Moles toluene converted}} = 1 - \frac{0.0036}{(1 - X)^{1.544}} \quad (j)$$

$$X \leq 0.97 \quad (k)$$

$$0.20 \leq \psi \leq 1.0 \quad (l)$$

Process Specifications

Production of benzene

80,000 metric ton per year (8000 = hour operation)

Molar feed

$$\frac{\text{Hydrogen}}{\text{Toluene}} = \frac{5 \text{ mol}}{1 \text{ mol}} \quad (m)$$

Objective Function

The objective function is to maximize profits, namely, products sold minus raw material costs. No capital or investment cost are involved in this example.

To prevent internal flow rates having zero shadow costs at the solution and therefore to avoid a multiplicity of solutions, a penalty of 0.05 times the price is incurred for each ton transported between a source and a destination.

Prices and Costs for Components

Component	Price/cost (\$/ton)
Toluene	200
Hydrogen	100
Benzene	500
Methane	100
Diphenyl	20

Phimister and colleagues obtained the optimal configuration and associated flow rates shown in Figure E14.6c using GAMS. The optimal value of the conversion was 0.697, and the selectivity ψ was 0.977, yielding a value of the objective function of \$18.65 million/year. Refer to Phimister and colleagues (1999) for a problem corresponding to a more complex plant involving four reactor–separator units and ten components.

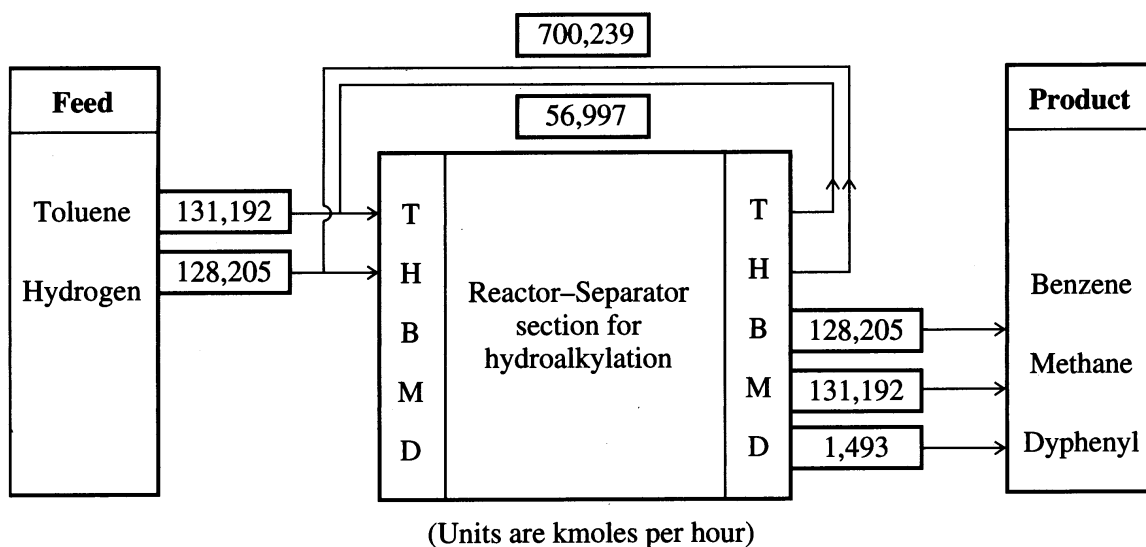


FIGURE E14.6c
Optimal configuration and stream flows.

REFERENCES

- Adjiman, C. S.; I. P. Androulakis; and C. A. Floudas. "A Global Optimization Method, α BB, for General Twice-Differentiable NLPs—II. Implementation and Computational Results." *Comput Chem Eng* **22**: 1159–1179 (1998).
- Androulakis, I. P.; G. D. Maranas; and C. A. Floudas. "Global Minimum Potential Energy Conformation of Oligo Peptides." *J Glob Opt* **11**: 1–34 (1997).
- Badgwell, T. A.; I. Trachtenberg; and T. F. Edgar. "Modeling and Scale-Up of Multiwafer LPCVD Reactors." *AIChE J* **38**(6): 926–938 (1992).
- Douglas, J. M. *Conceptual Design of Chemical Processes*. McGraw-Hill, New York (1988).
- Finlayson, B. *Nonlinear Analysis in Chemical Engineering*. McGraw-Hill, New York, 1980.
- Fogler, H. S. *Elements of Chemical Reaction Engineering*. Prentice-Hall, Upper Saddle River, NJ (1998).
- Froment, G. F.; and K. B. Bischoff. *Chemical Reactor Analysis and Design*. Wiley, New York (1990).
- Jensen, K. F.; and D. B. Graves. "Modeling and Analysis of Low Pressure CVD Reactors." *J Electrochem Soc* **130**: 1950–1957 (1983).
- Klepeis, J. L.; I. P. Androulakis; M. G. Ierapetritou, et al. "Predicting Solvated Peptide Configuration via Global Minimization of Energetic Atom-to-Atom Interactions." *Comput Chem Eng* **22**: 765–788 (1998).
- Levenspiel, O. *Chemical Reaction Engineering*. Wiley, New York (1998).
- Maranas, C. D.; and C. A. Floudas. "A Deterministic Global Optimization Approach for Molecular Structure Determination." *J Chem Phys* **100**: 1247–1261 (1994).
- Middleman, S.; and A. K. Hochberg. *Process Engineering Analysis in Semiconductor Device Fabrication*. McGraw-Hill, New York, 1993.
- Missen, R. W.; C. A. Mims; and B. A. Saville. *Introduction to Chemical Reaction Engineering and Kinetics*. Wiley, New York (1998).
- Murase, A.; H. L. Roberts; and A. O. Converse. "Optimal Thermal Design of an Autothermal Ammonia Synthesis Reactor." *Ind Eng Chem Process Des Dev* **9**: 503–513 (1970).
- Pardalos, P. M.; D. Shalloway; and G. Xue (editors). "Global Minimization of Nonconvex Energy Functions, Molecular Conformation and Protein Folding." DIMACS Series in Discrete Mathematics and Theoretical Computer Science, **23**, American Mathematical Society, Providence, RI (1996).
- Phimister, J. R.; E. S. Fragar; and J. W. Ponton. "The Synthesis of Multistep Process Plant Configurations." *Comput Chem Eng* **23**: 315–326 (1999).
- Roeningk, K. F.; and Jensen, K. F. "Analysis of Multicomponent LPCVD Processes." *J Electrochem Soc* **132**: 448–455 (1985).
- Sauer, R. N.; A. R. Coville; and C. W. Burwick. "Computer Points Way to More Profits." *Hydrocarbon Process Petrol Ref.* **43**: 84–92 (1964).
- Schlick, T.; R. D. Skeel; A. T. Brunger, et al. "Algorithmic Challenges in Computational Molecular Biophysics." *J Comput Phys* **151**: 9–48 (1999).
- Schmidt, L. D. *The Engineering of Chemical Reactions*. Oxford Univ. Press, Oxford (1997).
- Setalvad, T.; I. Trachtenberg; B. W. Bequette, et al. "Optimization of a Low Pressure CVD Reactor for the Deposition of Thin Films." *IEC Res* **28**: 1162–1172 (1989).
- Vasquez, M.; G. Nemethy; and H. A. Scheraga. "Conformational Energy Calculations of Polypeptides and Proteins." *Chem Rev* **94**: 2183–2239 (1994).

SUPPLEMENTARY REFERENCES

- Abel, O.; A. Helbig; W. Marquardt; H. Zwick, et al. "Productivity Optimization of an Industrial Semi-batch Polymerization Reactor Under Safety Constraints." *J Proc Contr* **10**: 351–362 (2000).
- Balakrishna, S.; and L. T. Biegler. "Targeting Strategies for the Synthesis and Energy Integration of Nonisothermal Reactor Networks." *Ind Eng Chem Res* **31**: 2152–2164 (1992).
- Bonvin, D. "Optimal Operation of Batch Reactors—A Personal View." *J Proc Contr* **8**: 355–368 (1998).
- Edwards, K.; V. Manousiouthakis; and T. F. Edgar. "Kinetic Model Reduction Using Genetic Algorithms." *Comput Chem Eng* **22**: 239–246 (1998).
- Feinberg, M.; and D. Hildebrandt. "Optimal Reactor Design from a Geometric Viewpoint: I. Universal Properties of the Attainable Region." *Chem Eng Sci* **52**(10): 1637–1666 (1997).
- Geddes, D.; and T. Kubera. "Integration of Planning and Real-Time Optimization Olefins Production." *Comput Chem Eng* **24**: 1645–1649 (2000).
- Guntern, C.; A. H. Keller; and K. Hungerbühler. "Economic Optimization of an Industrial Semi-batch Reactor Applying Dynamic Programming." *Ind Eng Chem Res* **37**(10): 4017–4022 (1998).
- Hildebrandt, D.; and L. T. Biegler. "Synthesis of Reactor Networks." In *Foundations of Computer Aided Process Design '94, AIChE Symposium Series*. L. T. Biegler; M. F. Doherty eds. **91**: 52–68 (1995).
- Kokossis, A. C.; and C. A. Floudas. "Optimization of Complex Reactor Networks—I. Isothermal Operation." *Chem Eng Sci* **45**(3): 595–614 (1990).
- Lakshmanan, A.; and L. T. Biegler. "Synthesis of Optimal Reactor Networks." *Ind Eng Chem Res* **35**(4): 1344–1353 (1996).
- Luus, R. "Optimization of Fed-batch Fermentors by Iterative Dynamic Programming." *Biotechnol Bioeng* **41**: 599–602 (1992).
- Rajesh, J. K.; S. K. Gupta; G. P. Rangaiiah; and A. K. Ray. "Multiobjective Optimization of Steam Reformer Performance Using Genetic Algorithm." *Ind Eng Chem Res* **39**(3): 706–717 (2000).
- Schweiger, C. A.; and C. A. Floudas. "Optimization Framework for the Synthesis of Chemical Reactor Networks." *Ind Eng Chem Res* **38**(3): 744–766 (1999).

**OPTIMIZATION IN LARGE-SCALE PLANT
DESIGN AND OPERATIONS**

15.1 Process Simulators and Optimization Codes 518
15.2 Optimization Using Equation-Based Process Simulators 525
15.3 Optimization Using Modular-Based Simulators 537
15.4 Summary 546
 References 546
 Supplementary References 548

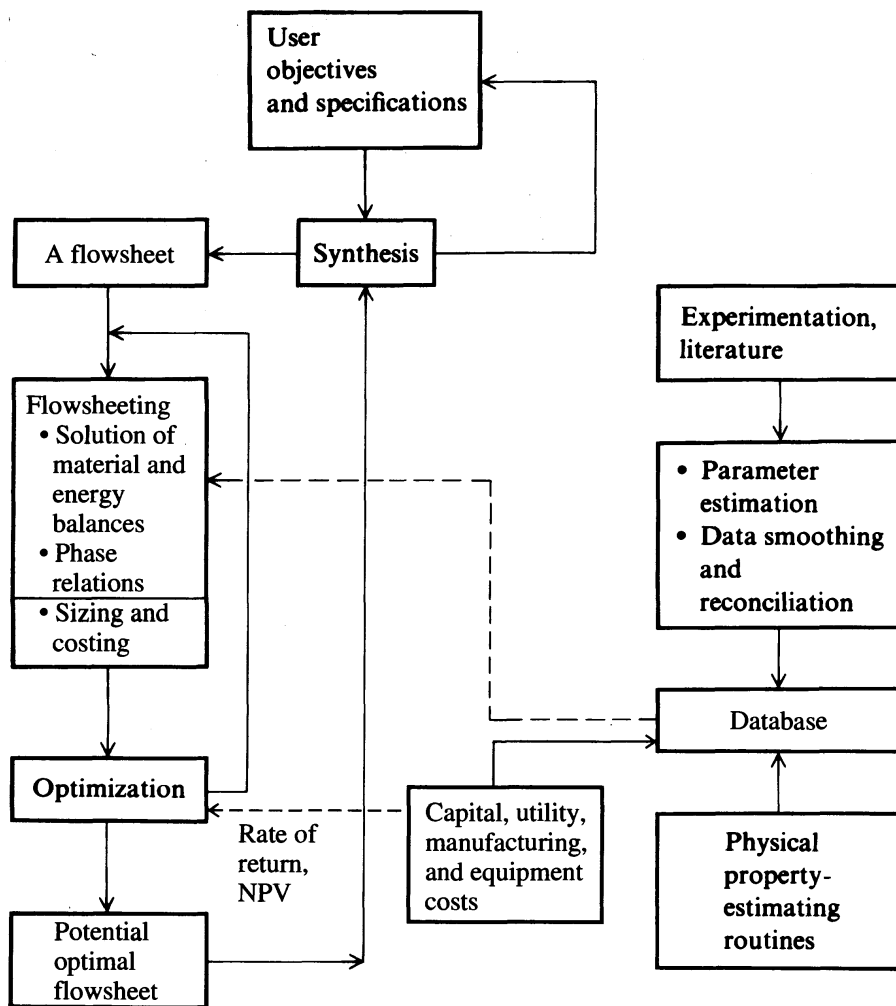


FIGURE 15.1
Information flow in the design process.

AS DISCUSSED IN Chapter 1, optimization of a large configuration of plant components can involve several levels of detail ranging from the most minute features of equipment design to the grand scale of international company operations. As an example of the size of the optimization problems solved in practice, Lowery et al. (1993) describe the optimization of a bisphenol-A plant via SQP involving 41,147 variables, 37,641 equations, 212 inequality constraints, and 289 plant measurements to identify the most profitable operating conditions. Perkins (1998) reviews the topic of plantwide optimization and its future.

An important global function of optimization is the synthesis of the optimal plant configuration (flowsheet). By *synthesis* we mean the designation of the structure of the plant elements, such as the unit operations and equipment, that will meet the designer's goals. Figure 15.1 shows the relation of synthesis to design and operation. You check a flowsheet for equipment that can be eliminated or rearranged, alternative separation methods, unnecessary feeds that can be eliminated, unwanted or hazardous product or byproducts that can be deleted, heat integration that can be improved, and so on. Even if no new technology is to be used, the problem is com-

binatorial in nature, and the number of alternatives increases substantially. For example, Gunderson and Grossman (1990) in synthesizing a heat exchanger network showed that for a net of five units below the pinch point (and three above), 126 different arrangements of exchangers exist. We have chosen not to discuss the general problem of synthesis in this chapter, but instead we treat examples of optimization applied to design of a specified configuration or flowsheet.

A major use of optimization is in the detailed *design* or retrofit of a plant for which the flowsheet is already formulated. Goals are to enhance profitability; reduce utility costs; select raw materials; size equipment; lay out piping; and analyze reliability, flexibility, and safety, and so on. Often, as a result of various case studies, a base case is developed by creating a detailed process flowsheet containing the major pieces of equipment. Then, process flow simulators are employed to achieve improved designs. The design team improves the database by getting vendor data and perhaps pilot plant data; simulates the base case design to find improvements and barriers to feasibility; and develops networks of heat exchangers, turbines, and compressors to satisfy the heating, cooling, and power requirements of the process. Refer to any of the process design books such as Seider et al. (1999) for details concerning the design process.

An even more widespread application of optimization is the determination of the optimal operating conditions for an existing plant, such as selecting particular feedstocks, temperatures, pressures, flow rates, and so on. Figure 15.2 traces the information flow involved in determining the optimal plant operating conditions. See Chapter 16 for a discussion of the optimization hierarchy in plant operations. As indicated in the figure, optimization occurs at intermediate stages of the process simulator as well as in the overall economic evaluation. The figure implies that effectively meshing optimization algorithms with process simulators requires more than just an optimization code and a process simulator containing the process model. The software functions involved are

1. A supervisor or director to manage overall control of the software components.
2. Data processing conditioning, reconciliation, and validation of the data evolving from the plant.
3. Estimation of process parameters and unmeasured variables.
4. Optimization of different kinds of problems.
5. Simulation of plant models (equations, modules, or both) of varying degrees of detail.
6. A database (historian) for process variables, costs and revenues, operating conditions, disturbances, and so on.
7. Communication links for data transfer and command signals.
8. Reports and analysis capability for unit and plant performance, economic performance, and hypothetical scenarios.

Although uncertainty exists in the results of all cases of the optimization of plants because of the uncertainty in the values of the parameters in the process models themselves, in the cost and revenue values in the objective function, and in potential changes in the process inputs, we avoid such issues in this chapter and focus solely on deterministic optimization.

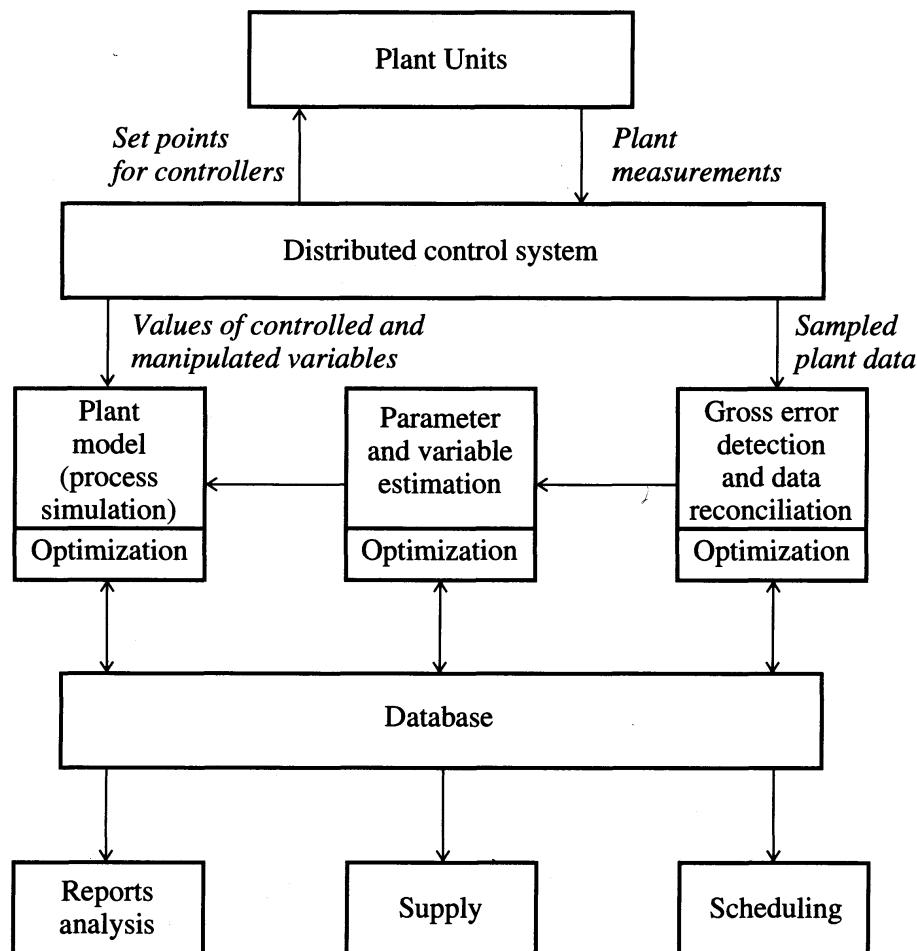


FIGURE 15.2
Information flow in developing the optimal operating conditions.

15.1 PROCESS SIMULATORS AND OPTIMIZATION CODES

Process simulators contain the *model* of the process and thus contain the bulk of the constraints in an optimization problem. The equality constraints (“hard constraints”) include all the mathematical relations that constitute the material and energy balances, the rate equations, the phase relations, the controls, connecting variables, and methods of computing the physical properties used in any of the relations in the model. The inequality constraints (“soft constraints”) include material flow limits; maximum heat exchanger areas; pressure, temperature, and concentration upper and lower bounds; environmental stipulations; vessel hold-ups; safety constraints; and so on. A *module* is a model of an individual element in a flowsheet (e.g., a reactor) that can be coded, analyzed, debugged, and interpreted by itself. Examine Figure 15.3a and b.

Two extremes are encountered in process simulator software. At one extreme the process model comprises a set of equations (and inequalities) so that the process model equations form the constraints for optimization, exactly the same as described in previous chapters in this book. This representation is known as an *equation-*

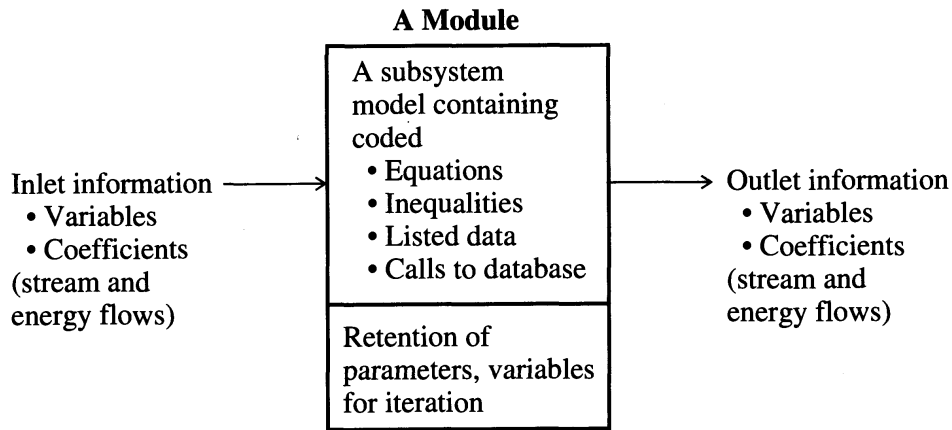


FIGURE 15.3a
A typical process module showing the necessary interconnections of information.

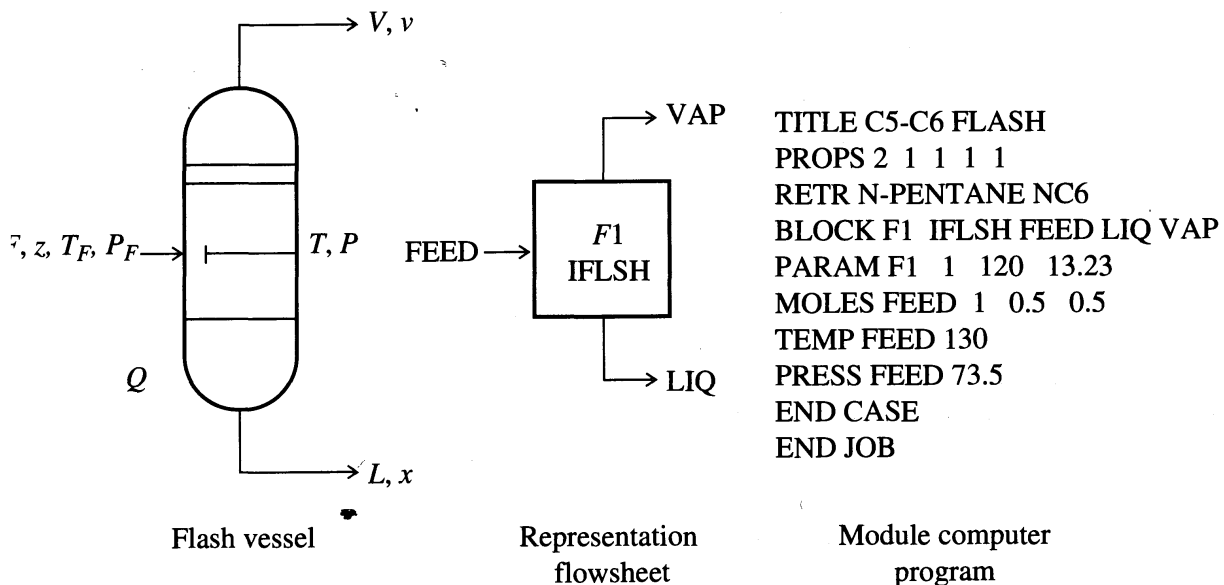


FIGURE 15.3b
A module that represents a flash unit. (Reproduced, with permission, from J. D. Seader, W. D. Seider, and A. C. Pauls. *Flowtran Simulation—An Introduction*. Austin, TX: CACHE, 1987.)

oriented process simulator. The equations can be solved in a sequential fashion analogous to the modular representation described in the next section, or simultaneously by Newton's method or by employing sparse matrix techniques to reduce the extent of matrix manipulations (Gill et al., 1981). Two of the better known equation-based codes are Aspen Custom Modeler (Aspen Technology 1998) and ASCEND (Westerberg 1998). Equation-based codes such as DMCC and RT-OPT (Aspen Technology), and ROMEO (Simulation Sciences, 1999) dominate closed-loop, real-time optimization applications (refer to Chapter 16). Section 15.2 covers meshing equation-based process simulators with optimization algorithms.

At the other extreme, the process can be represented on a flowsheet by a collection of modules (a modular-based process simulator) in which the equations (and other information) representing each subsystem or piece of equipment are coded so that a module may be used in isolation from the rest of the flowsheet and hence is portable from one flowsheet to another. Each module contains the equipment sizes, the material and energy balance relations, the component flow rates, temperatures, concentrations, pressures, and phase conditions. Examples of commercial codes are ASPEN PLUS (Aspen Technology, 1998), HYSYS (Hyprotech, 1998), ChemCAD (Chemstations, 1998), PRO/II 1998 (Simulation Sciences, 1998), and Batch Pro and Enviro Pro Designer (Intelligen, 1999). Section 15.3 covers meshing modular-based process simulators with optimization algorithms.

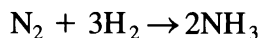
In addition to the two extremes, combinations of equations and modules can be used. Equations can be lumped into modules, and modules can be represented by their basic equations or by polynomials that fit the input–output information.

Although, as explained in Chapter 9, many optimization problems can be naturally formulated as mixed-integer programming problems, in this chapter we will consider only steady-state nonlinear programming problems in which the variables are continuous. In some cases it may be feasible to use binary variables (on–off) to include or exclude specific stream flows, alternative flowsheet topography, or different parameters. In the economic evaluation of processes, in design, or in control, usually only a few (5–50) variables are decision, or independent, variables amid a multitude of dependent variables (hundreds or thousands). The number of dependent variables in principle (but not necessarily in practice) is equivalent to the number of independent equality constraints plus the active inequality constraints in a process. The number of independent (decision) variables comprises the remaining set of variables whose values are unknown. Introduction into the model of a specification of the value of a variable, such as $T = 400^{\circ}\text{C}$, is equivalent to the solution of an independent equation and reduces the total number of variables whose values are unknown by one.

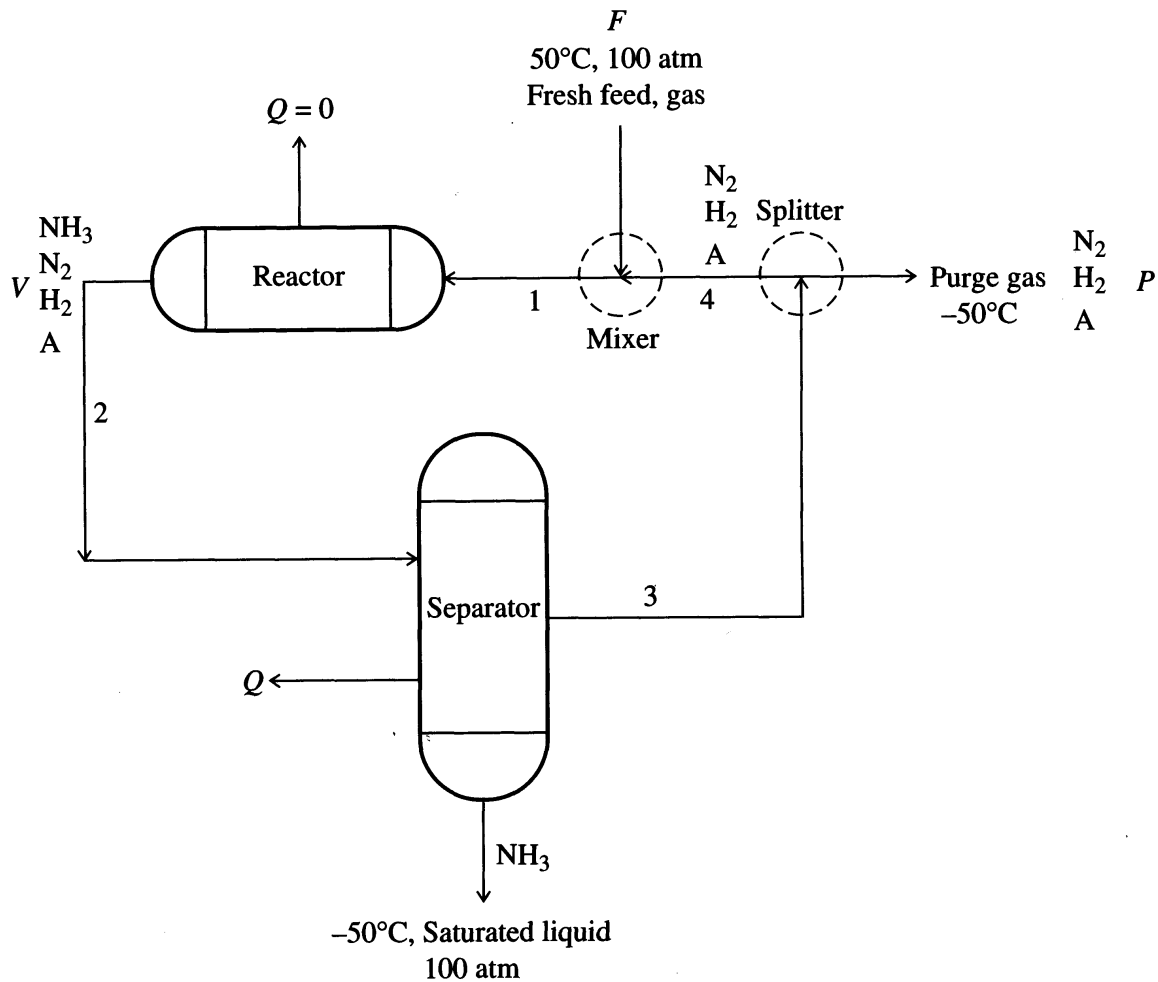
In optimization using a process simulator to represent the model of the process, the *degrees of freedom* are the number of decision variables (independent variables) whose values are to be determined by the optimization, hence the results of an optimization yield a fully determined set of variables, both independent and dependent. Chapter 2 discussed the concept of the degrees of freedom. Example 15.1 demonstrates the identification of the degrees of freedom in a small process.

EXAMPLE 15.1 CALCULATION OF THE DEGREES OF FREEDOM

Figure E15.1 shows a simplified flowsheet for the conversion of N_2 and H_2 to ammonia (NH_3) when argon (A) is present in the feed. After the reaction of N_2 and H_2



the NH_3 is separated as a liquid from the gas phase. A purge gas stream prevents argon build-up in the system. Fresh feed is introduced in the proper ratio of N_2 to H_2 with


FIGURE E15.1

the accompanying argon of about 0.9 percent. Assume all of the units and the pipe lines are adiabatic ($Q = 0$). The fraction conversion in the reactor is 25 percent.

The process has four separate subsystems for the degree-of-freedom analysis. Redundant variables and redundant constraints are removed to obtain the net degrees of freedom for the overall process. The 2 added to N_{sp} refers to the conditions of temperature and pressure in a stream; +1 represents the heat transfer Q .

In this example

N_v is the number of variables

N_{sp} is the number of components (species) in a stream

N_r is the number of independent constraints

N_d is the degrees of freedom (number of decision variables)

The analysis of each subsystem is as follows.

Mixer:

$$N_v = 3(N_{sp} + 2) + 1 = 3(6) + 1 = 19$$

N_r :

Material balances (H_2 , N_2 , A only) 3

Energy balance 1

Specifications:

NH ₃ concentration is zero	3	
$T_P = -50^\circ\text{C}$	1	
$T_F = 50^\circ\text{C}$	1	
Assume that $p_F = p_{\text{mix out}} = p_{\text{split}} = 100$	3	
$Q = 0$	<u>1</u>	<u>13</u>
$N_d: 19 - 13 =$		<u>6</u>

Reactor:

$$N_v = 2(N_{sp} + 2) + 1 = 2(6) + 1 = 13$$

 $N_r:$

Material balances (H, N, A) 3

Energy balances 1

Specifications:NH₃ entering = 0 1 $Q = 0$ 1

Fraction conversion 1

 $p_{\text{in}} = p_{\text{out}} = 100 \text{ atm}$ 2Energy balance 1 10

$$N_d: 13 - 10 = \underline{\underline{3}}$$

Separator:

$$N_v = 3(N_{sp} + 2) + 1 = 3(6) + 1 = 19$$

 $N_r:$

Material balances 4

Energy balance 1

Specifications: $T_{\text{out}} = -50^\circ\text{C}$ 1 $p_r = p_{\text{in}} = p_{\text{NH}_3} = 100$ 3NH₃ concentration is 0 in
recycle gas 1N₂, H₂, A are 0 in liquid NH₃ 3 13

$$N_d: 19 - 13 = \underline{\underline{6}}$$

Splitter:

$$N_v = 3(N_{sp} + 2) = 3(6) = 18$$

 $N_r:$

Material balances 1

Specifications:NH₃ concentration = 0 1Compositions same $2(N_{sp} - 1)$ 6Stream temperatures same = -50°C 3Stream pressures same = 100 atm 3 14

$$N_d: 18 - 14 = \underline{\underline{4}}$$

The total number of degrees of freedom is 19 less the redundant information, which is as follows:

Redundant variables in interconnecting streams being eliminated:

$$\text{Stream 1: } (4 + 2) = 6$$

$$\text{Stream 2: } (4 + 2) = 6$$

$$\text{Stream 3: } (4 + 2) = 6$$

$$\text{Stream 4: } (4 + 2) = \underline{6}$$

24

Redundant constraints being eliminated:

Stream 1:

$$\text{NH}_3 \text{ concentration} = 0 \quad 1$$

$$p = 100 \text{ atm} \quad 1$$

Stream 2:

$$p = 100 \text{ atm} \quad 1$$

Stream 3:

$$\text{NH}_3 \text{ concentration} = 0 \quad 1$$

$$p = 100 \text{ atm} \quad 1$$

$$T = -50^\circ\text{C} \quad 1$$

Stream 4:

$$\text{NH}_3 \text{ concentration} = 0 \quad 1$$

$$T = -50^\circ\text{C} \quad 1$$

$$p = 100 \text{ atm} \quad \underline{1}$$

9

Overall the number of degrees of freedom should be

$$N_d = 19 - 24 + 9 = 4$$

The redundant constraints and variables can be regarded as $24 + 9 = 33$ additional equality constraints in the optimization problem.

In optimization using a modular process simulator, certain restrictions apply on the choice of decision variables. For example, if the location of column feeds, draws, and heat exchangers are selected as decision variables, the rate or heat duty cannot also be selected. For an isothermal flash both the temperatures and pressure may be optimized, but for an adiabatic flash, on the other hand, the temperature is calculated in a module and only the pressure can be optimized. You also have to take care that the decision (optimization) variables in one unit are not varied by another unit. In some instances, you can make alternative specifications of the decision variables that result in the same optimal solution, but require substantially different computation time. For example, the simplest specification for a splitter would be a molar rate or ratio. A specification of the weight rate of a component in an exit flow stream from the splitter increases the computation time but yields the same solution.

Next, we need to clarify some of the jargon that you will find in the literature and documentation associated with commercial codes that involve process simulators. Two major types of optimization algorithms exist for nonlinear programming.

1. *Feasible path algorithms*. The equality constraints and active inequality constraints are satisfied at the end of every intermediate stage of the calculations.
2. *Infeasible path algorithms*. The equality constraints and active inequality constraints are satisfied only at the stage on which the optimal solution is reached.

Clearly option 1 incurs more computation time when process simulators are involved, but an abnormal termination yields a feasible solution.

Another classification of optimization codes relates whether a full set of variables is used in the search:

1. *Full vector*. All the independent and dependent variables constitute the vector of variables in the search.
2. *Reduced vector*. Only the independent variables are involved in the search; the dependent variables are then determined from the constraints.

With respect to process simulators, we can identify three types, with hybrid types often occurring:

1. *Equation-based*. Explained previously.
2. *Sequential modular*. Refers to the process simulator being based on modules, and the modules solved in a sequential precedence order imposed by the flow-sheet information flow.
3. *Simultaneous modular*. The process simulator is composed of modules, but simplified, approximate, or partial representation of the modules enables solution techniques used in equation-based methods to be employed.

Other jargon you will encounter:

1. *Online*. Optimization calculations are carried out by computers that process plant data and transmit control signals.
2. *Offline*. Data is collected and used subsequently by separate computers for optimization so that the results are not directly available.
3. *Real time*. The clock cycle for the collection and transfer of process data and the optimization calculations is the same.

The kinds of optimization codes most often used together with process simulators include

1. Linear programming: LP (refer to Chapter 7).
2. Sequential linear programming: SLP (refer to Chapter 8).
3. Sequential quadratic programming: SQP (refer to Chapter 8).
4. Generalized reduced gradient: GRG (refer to Chapter 8).
5. Nonlinear programming: NLP—other than items 3 or 4 (refer to Chapter 8).
6. Mixed-integer nonlinear programming: MINLP (refer to Chapter 9).
7. Mixed-integer successive quadratic programming (refer to Chapter 9).
8. Random search (refer to Chapter 10 or Section 6.1).

Commercial process simulators mainly use a form of SQP. To use LP, you must balance the nonlinearity of the plant model (constraints) and the objective function with the error in approximation of the plant by linear models. Infeasible path, sequential modular SQP has proven particularly effective.

Finally, we should mention that in addition to solving an optimization problem with the aid of a process simulator, you frequently need to find the sensitivity of the variables and functions at the optimal solution to changes in fixed parameters, such as thermodynamic, transport and kinetic coefficients, and changes in variables such as feed rates, and in costs and prices used in the objective function. Fiacco in 1976 showed how to develop the sensitivity relations based on the Kuhn–Tucker conditions (refer to Chapter 8). For optimization using equation-based simulators, the sensitivity coefficients such as $(\partial h_i / \partial x_i)$ and $(\partial x_i / \partial x_j)$ can be obtained directly from the equations in the process model. For optimization based on modular process simulators, refer to Section 15.3. In general, sensitivity analysis relies on linearization of functions, and the sensitivity coefficients may not be valid for large changes in parameters or variables from the optimal solution.

15.2 OPTIMIZATION USING EQUATION-BASED PROCESS SIMULATORS

In this section we consider general process simulator codes rather than specialized codes that apply only to one plant. To mesh equation-based process simulators with optimization codes, a number of special features not mentioned in Chapter 8 must be implemented.

1. A method of formatting the equations and inequality constraints. Slack variables are used to transform the inequality constraints into equality constraints.
2. A possibility of using both continuous and discrete variables, the latter being particularly necessary to accommodate changes in phase or changes from one correlation to another.
3. The option of using alternative forms of a function depending on the value of logical variables that identify the state of the process. Typical examples are the shift in the relations used to calculate the friction factor from laminar to turbulent flow, or the calculation of $P - V - T$ relations as the phase changes from gas to liquid.
4. Efficient methods for solving equations in the physical property database (which often require up to 80% of the computation time needed to solve a plant optimization problem).
5. Efficient methods for solving large sets of linear equations, for example, the linearized constraints, particularly involving sparse matrices.
6. A good method of selecting initial guesses for the solution of the algebraic equations. Poor choices lead to unsatisfactory results. You want the initial guesses to be as close to the optimal solution as possible so that the procedure will converge, and converge rapidly. We recommend running the process simulator alone to develop one or more base cases that will serve feasible starting points for the optimization.

7. Provision for *scaling* of the variables and equations. By scaling variables we mean introducing transformations that make all the variables have ranges of the same order of magnitude. By scaling of equations we mean multiplying each equation by a factor that causes the value of the deviation of each equation from zero to be of the same order of magnitude. User interaction and analysis for a specific problem is one way to introduce scaling.
8. The code must carry out a structural analysis to determine if the model is well posed, that is, can it detect any inconsistencies among the equations in the model (Duff et al., 1989; Zaher, 1995)?

Figure 15.4 shows how the nonlinear optimization problem fits in with two widely used optimization algorithms: the generalized reduced gradient (GRG) and successive quadratic programming (SQP). The notation is in Table 15.1. Slack variables \mathbf{x}_s have been added to the inequality constraints $\mathbf{g} \geq \mathbf{0}$ to convert them to equality constraints. The formulation in Figure 15.4 assumes that the functions and variables are continuous and differentiable (in practice, finite differences may be used as substitutes for analytical derivatives). Although we will not discuss optimization of dynamic processes in this chapter, in the NLP problem you can insert differential equations as additional equality constraints. Refer to Ramirez (1994) for details. In the execution of the optimization code, in some phases the specific assignment of independent and dependent variables within the code may differ from those you designate.

In formatting the inequalities g and equations h , you will find that the so-called open-equation representation is preferred to the closed-equation representation. One of the simplest examples is a heat exchanger model (the closed-equation format):

$$Q = F_C C_{pC} (T_{C,out} - T_{C,in})$$

$$Q = F_H C_{pH} (T_{H,in} - T_{H,out})$$

$$Q = UA \left[\frac{(T_{H,in} - T_{C,out}) - (T_{H,out} - T_{C,in})}{\ln \left[\frac{(T_{H,in} - T_{C,out})}{(T_{H,out} - T_{C,in})} \right]} \right]$$

where A = heat transfer area

C_p = heat capacity

F = flow rate

Q = heat transferred

T = temperature

U = heat transfer coefficient

H = hot

C = cold

If the temperatures, heat capacities, U , and A are known quantities, then you can directly calculate Q and the F 's. On the other hand, if you know the stream flows,

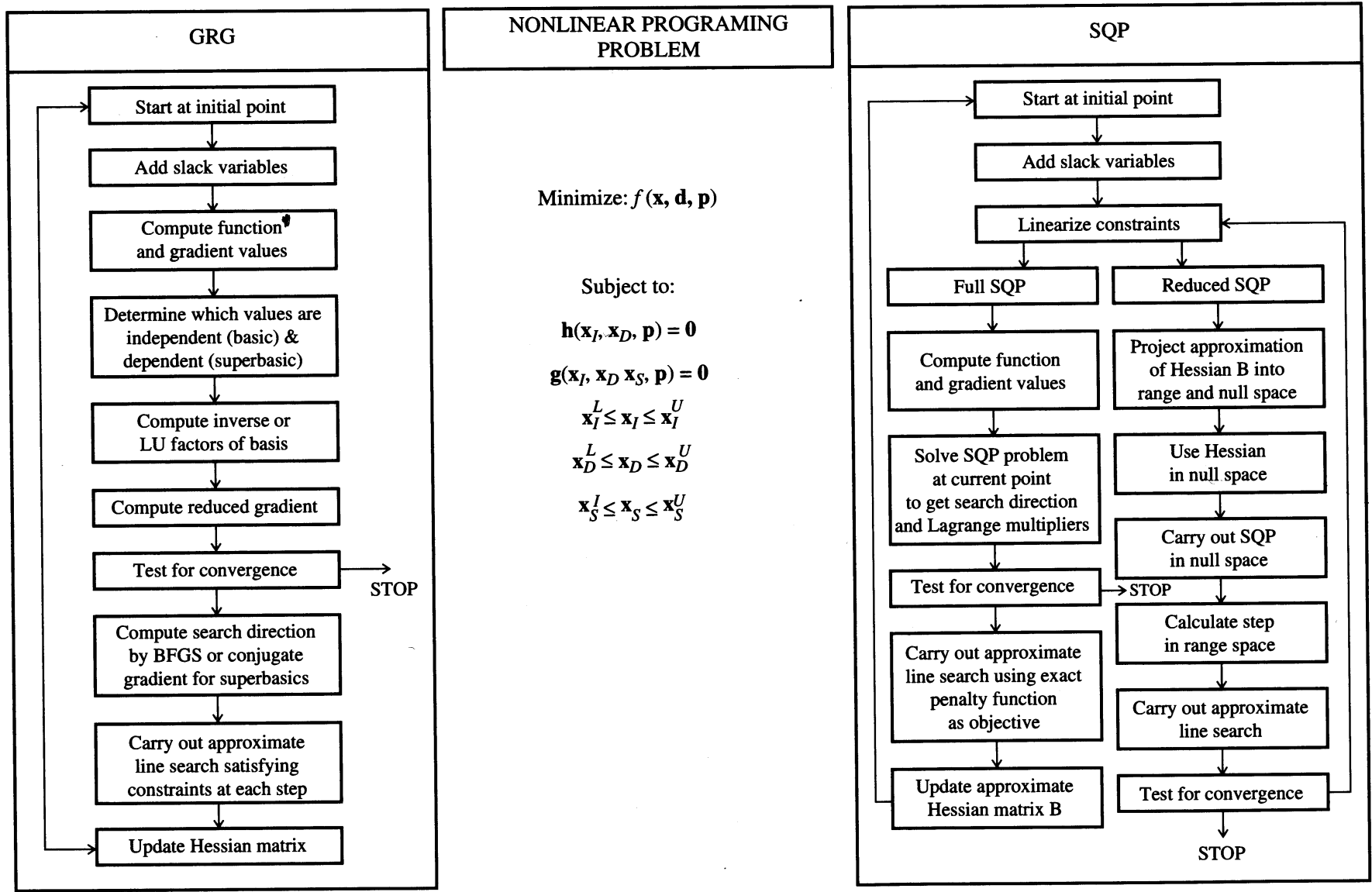


FIGURE 15.4

TABLE 15.1
Notation for Figure 15.4

f	Objective function
\mathbf{g}	Set of inequality constraints
\mathbf{h}	Set of equality constraints
\mathbf{p}	Vector of coefficients in the objective function and constraints
\mathbf{x}_I	Vector of independent (decision) variables
\mathbf{x}_D	Vector of dependent variables
\mathbf{x}_S	Vector of slack variables added to the inequality constraints
L	lower bound
U	upper bound

inlet temperatures, C_p 's, U , and A , then the solution for Q and the outlet temperatures must be determined via iteration. This problem arises particularly for process models in which one unit that is underspecified is connected with another unit that is overspecified.

By using open-equation formats and infeasible path optimization algorithms, the type of difficulty described above can be avoided. All the equations in the NLP problem can be solved simultaneously, driving the residuals to zero. The open-equation format for the heat exchanger is

$$\begin{aligned}
 R_1 &= Q - F_C C_{pC} (T_{C,out} - T_{C,in}) \\
 R_2 &= Q - F_H C_{pH} (T_{H,in} - T_{H,out}) \\
 R_3 &= Q \ln \left[\frac{(T_{H,in} - T_{C,out})}{(T_{H,out} - T_{C,in})} \right] - UA [(T_{H,in} - T_{C,out}) - (T_{H,out} - T_{C,in})]
 \end{aligned}$$

where R_i is a residual. Note that division by a logarithm has been eliminated.

Another advantage of the open-equation format is that simple connection equations can be used rather than eliminating variables and equations that are connected. For example, the connections between two heat exchangers can be formulated as

$$\begin{aligned}
 R_1 &= F_{C,out,1} - F_{C,in,2} \\
 R_2 &= T_{C,out,1} - T_{C,in,2} \\
 R_3 &= F_{H,out,2} - F_{H,in,1} \\
 R_4 &= T_{H,out,2} - T_{H,in,1}
 \end{aligned}$$

More variables are retained in this type of NLP problem formulation, but you can take advantage of sparse matrix routines that factor the linear (and linearized) equations efficiently. Figure 15.5 illustrates the sparsity of the Hessian matrix used in the QP subproblem that is part of the execution of an optimization of a plant involving five unit operations.

Figure 15.4 shows the Hessian matrix for two different types of SQP algorithms for solving large-scale optimization problems. In the full-space SQP, all of

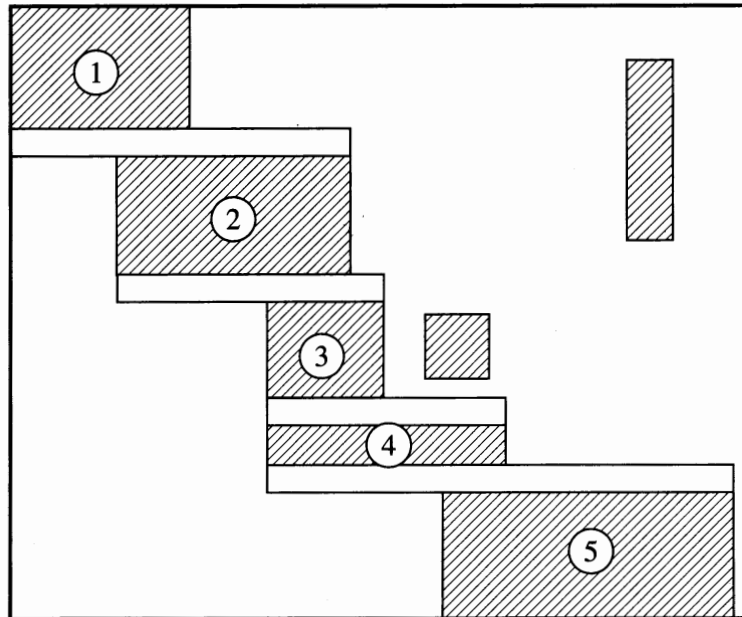


FIGURE 15.5

The Hessian matrix for the QP subproblem showing five units and the sparsity of the matrix.

the variables, both independent and dependent, are solved for simultaneously in the set of linear(ized) equations in the QP subproblem. The sparse structure of both \mathbf{B} and $\Delta\mathbf{h}$ can be taken advantage of in their solution. In the reduced-space SQP only the sparse structure of $\Delta\mathbf{h}$ is used. For specific details of the execution of the reduced SQP, refer to the summary in Biegler et al. (1997) and the references therein, and to Schmid and Biegler (1994a).

As mentioned before, two contrasting classes of strategies exist for executing the SQP algorithms:

1. Feasible path strategies.
2. Infeasible path strategies.

With feasible path strategies, as the name implies, on each iteration you satisfy the equality and inequality constraints. The results of each iteration, therefore, provide a candidate design or feasible set of operating conditions for the plant, that is, sub-optimal. Infeasible path strategies, on the other hand, do not require exact solution of the constraints on each iteration. Thus, if an infeasible path method fails, the solution at termination may be of little value. Only at the optimal solution will you satisfy the constraints.

To improve the formatting of the equations that represent a plant, many commercial codes partition the equations into groups of irreducible sets of equations, that is, those that have to be solved simultaneously. If a plant is represented by thousands of equations, the overall time consumed in their solution via either a GRG or SQP algorithm is reduced by partitioning and rearranging the order of the equations with the result indicated in Figure 15.6. Organization of the set of equations into irreducible sets can be carried out by the use of permutation matrices or by one of

$$\begin{aligned}
 h_1: & x_1^2 x_2 - 2x_3^{1.5} + 4 = 0 \\
 h_2: & x_2 + 2x_5 - 8 = 0 \\
 h_3: & x_1 x_4 x_5^2 - 2x_3 - 7 = 0 \\
 h_4: & -2x_2 + x_5 + 5 = 0 \\
 h_5: & x_2 x_4^2 x_5 + x_2 x_4 - 6 = 0
 \end{aligned}$$

(a) The n independent equations involving n variables ($n = 5$).

	x_1	x_2	x_3	x_4	x_5
h_1	1	1	1		
h_2		1			1
h_3	1		1	1	1
h_4		1			1
h_5		1		1	1

(b) The occurrence matrix (the 1's represent the occurrence of a variable in an equation).

	x_2	x_5	x_4	x_1	x_3
h_2	1	1			
h_4	1	1	I		
h_5	1	1	1		
h_3		1			III
h_1	1				1

(c) The rearranged (partitioned) occurrence matrix with groups of equations (sets I, II, and III) that have to be solved simultaneously collected together in the precedence order for solution.

FIGURE 15.6

Partitioning of sets of independent equations increases the sparsity of the occurrence matrix.

the many algorithms found in Himmelblau (1973). Feedback of information, materials, or energy ties equations together in irreducible groups.

We next solve an example optimization problem for a plant represented by equations and inequalities using the GRG method.

EXAMPLE 15.2 PROCESS OPTIMIZATION VIA GRG (EQUATION-BASED SOFTWARE)

Figure E15.2 shows the flowsheet for the process. Feed (stream 1) is a vapor mixture of ethane, propane, and butane (in the proportions shown in the figure) at 200°F and 500 psia. The product stream (stream 8) is a liquid at $\leq -20^\circ\text{F}$ having the same composition but a reduced pressure. The notation for this example is defined in Table E15.2A.

1 Objective function. A simple objective function is used, namely, the minimization of the instantaneous cost of the work done by the three recycle compressors:

$$\text{Minimize: } f = C \left[F_3 \frac{(H_9 - H_3)}{0.65} + F_5 \frac{(H_{11} - H_5)}{0.65} + F_7 \frac{(H_{13} - H_7)}{0.65} \right] \quad (a)$$

The value 0.65 is the efficiency factor.

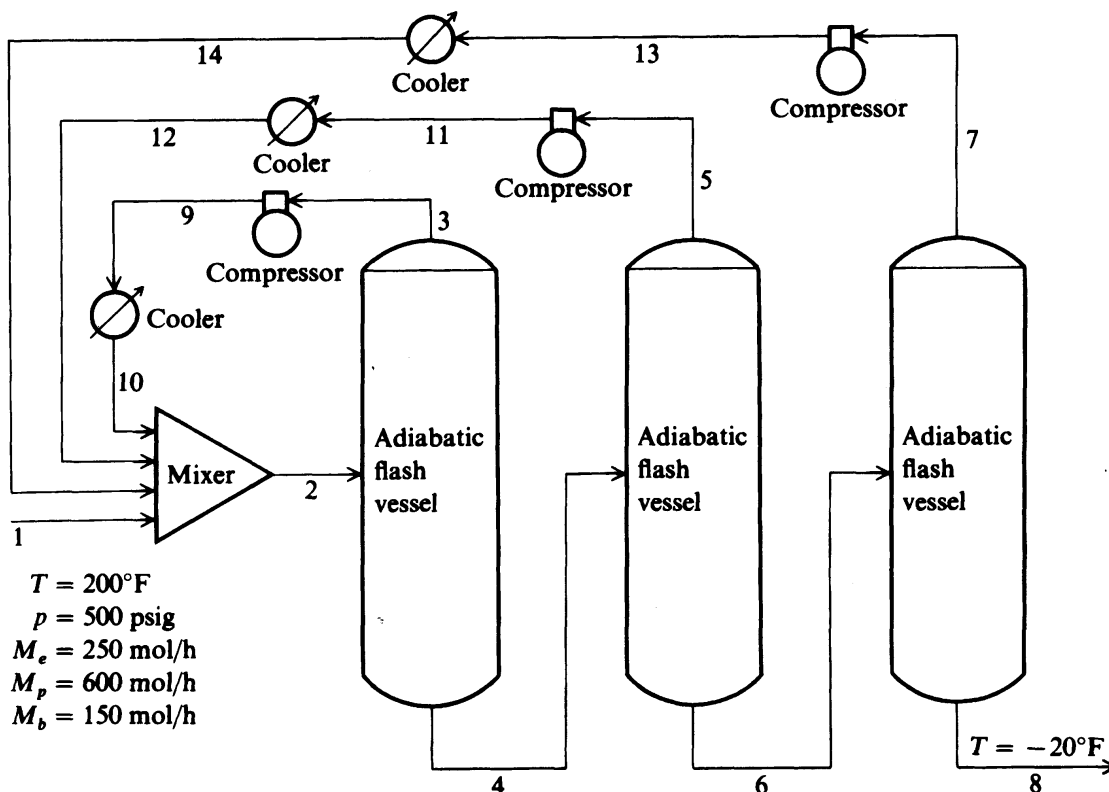


FIGURE E15.2
Flow diagram of light hydrocarbon refrigeration process.

TABLE E15.2A
Notation for Example 15.2

C	A constant denoting the cost of work per unit energy
F_j	Total molar flow rate of process stream j
H_j	Molar enthalpy of process stream j
K_i	Vapor-liquid equilibrium constant for component i
L_j	Liquid molar flow rate of process stream j
p_i	Pressure of process stream identified by subscript ($p_4 = p_3$, $p_6 = p_5$, and $p_8 = p_7$)
T_i	Temperature of process stream identified by subscript ($T_4 = T_3$, $T_6 = T_5$, and $T_8 = T_7$)
V_j	Vapor molar flow rate of process stream j
$x_{i,j}$	Liquid molar flow rate of component i in process stream j [$i = 1$ (ethane), 2 (propane), 3 (n -butane)]
$y_{i,j}$	Vapor molar flow rate of component i in process stream j [$i = 1$ (ethane), 2 (propane), 3 (n -butane)]

2 Inequality constraints. Three inequality constraints are involved: two relating pressures and one product temperature specification.

$$p_5 - p_3 \leq 0 \quad (b-1)$$

$$p_7 - p_5 \leq 0 \quad (b-2)$$

$$T_7 + 20 \leq 0 \quad (b-3)$$

In addition, all 34 values of T_j , p_j , $x_{i,j}$, and $y_{i,j}$ have lower and upper bounds.

3 Equality constraints. The equality constraints (30 in all) are the linear and nonlinear material and energy balances and the phase relations.

3.1 Mixer.

Material balances:

$$\frac{0.5(y_{i,1} + x_{i,10} + x_{i,12} + x_{i,14} - y_{i,2} - x_{i,2})}{\max \{1, (y_{i,1} + x_{i,10} + x_{i,12} + x_{i,14} + y_{i,2} + x_{i,2})/2\}} = 0, \quad i = 1, \dots, 3 \quad (c)$$

Energy balance:

$$\frac{0.05(F_1H_1 + F_{10}H_{10} + F_{12}H_{12} + F_{14}H_{14} - F_2H_2)}{\max \{1, (F_1H_1 + F_{10}H_{10} + F_{12}H_{12} + F_{14}H_{14})/2\}} = 0 \quad (d)$$

The denominators in this example are simply scaling factors in the respective constraints evaluated using the values of the variables in the numerator; 1 or the other term is picked, whichever is bigger. For example, the terms in the denominators of Equations (c) and (d) representing the average of the mass and energy, respectively, in and out, as well as the denominators of the following equations, are not needed for the balances—they are scaling factors (as are the multipliers 0.05 or 0.5) that are introduced to improve the conditioning of the matrices of partial derivatives of the constraints. Without such scaling, the non-linear programming code may not reach the optimal solution but instead terminate prematurely.

3.2 Adiabatic flash vessels.

Material balances:

$$\frac{0.5(y_{i,2} + x_{i,2} - y_{i,3} - x_{i,4})}{\max \{1, (y_{i,2} + x_{i,2} + y_{i,3} + x_{i,4})/2\}} = 0, \quad i = 1, \dots, 3 \quad (e)$$

$$\frac{0.5(x_{i,4} - y_{i,5} - x_{i,6})}{\max \{1, (x_{i,4} + y_{i,5} + x_{i,6})/2\}} = 0, \quad i = 1, \dots, 3 \quad (f)$$

$$\frac{0.5(x_{i,6} + y_{i,7} - x_{i,8})}{\max \{1, (x_{i,6} + y_{i,7} + x_{i,8})/2\}} = 0, \quad i = 1, \dots, 3 \quad (g)$$

Energy balances: In the energy balances the multiplier 0.05 is used to assist in scaling.

$$\frac{0.05(F_2H_2 - F_3H_3 - F_4H_4)}{\max\{1, (F_2H_2 + F_3H_3 + F_4H_4)/2\}} = 0$$

$$\frac{0.05(F_4H_4 - F_5H_5 - F_6H_6)}{\max\{1, (F_4H_4 + F_5H_5 + F_6H_6)/2\}} = 0$$

$$\frac{0.05(F_6H_6 - F_7H_7 - F_8H_8)}{\max\{1, (F_6H_6 + F_7H_7 + F_8H_8)/2\}} = 0$$

The values for the enthalpies of the streams in the database were based on the Curl–Pitzer correlations (Green, 1997). The enthalpies are calculated from correlations at zero pressure (functions of temperature and composition only) and then corrected via the enthalpy deviation:

$$H = H^0 - \left(\frac{H^0 - H}{T_c}\right)T_c \quad (h)$$

where H^0 is the stream molar enthalpy and the superscript 0 designates zero pressure, and T_c is the critical temperature. The enthalpy deviation term itself, $\Delta H/T_c$, is a function of the mole weighted average of the three critical properties: temperature, pressure, and compressibility.

3.3 *Energy balances for compressors.* For isentropic compression

$$\frac{0.05[T_9 - (T_3 + 459.69)(500.0/P_3)^{0.200} - 459.69]}{\max\{1, (T_3 + T_9)/2\}} = 0 \quad (i)$$

$$\frac{0.05[T_{11} - (T_5 + 459.69)(500.0/P_5)^{0.200} - 459.69]}{\max\{1, (T_5 + T_{11})/2\}} = 0 \quad (j)$$

$$\frac{0.05[T_{13} - (T_7 + 459.69)(500.0/P_7)^{0.200} - 459.69]}{\max\{1, (T_7 + T_{13})/2\}} = 0 \quad (k)$$

3.4 *Phase equilibria relations.* Evaluation of the K values for phase equilibria was based on the relation

$$K_i = \frac{\gamma_{il}\nu_i}{\phi_i} \quad (l)$$

where γ_{il} = activity coefficient in the liquid phase of component i evaluated from Hildebrand and Scott (Green, 1997)

ν_i = fugacity coefficient of component i in the liquid phase evaluated from Chao-Seader (Green, 1997)

ϕ_i = fugacity coefficient of component i in the vapor phase evaluated from Redlich-Kwong (Green, 1997)

Based on the notation of Table E15.2a, in stream j

$$K_i = \frac{y_{i,j-1}/V_j}{x_{i,j}/L_j} \quad (m)$$

To assist in scaling, Equation (m) is rearranged as follows:

$$x_{i,j}K_i\left(\frac{V_j}{L_j}\right) + x_{i,j} = y_{i,j-1} + x_{i,j}$$

$$x_{i,j} = \frac{(x_{i,j} + y_{i,j-1})L_j}{K_iV_j + L_j}$$

or

$$x_{i,j} - \frac{(x_{i,j} + y_{i,j-1})L_j}{K_iV_j + L_j} = 0$$

and divided by

$$\max\left\{1, x_{i,j} + \frac{(x_{i,j} + y_{i,j-1})L_j}{K_iV_j + L_j}\right\}$$

and multiplied by the factor 0.01:

$$\frac{0.01\{x_{i,2} - [(x_{i,2} + y_{i,2})L_2/(K_iV_2 + L_2)]\}}{\max\{1, x_{i,2} + [(x_{i,2} + y_{i,2})L_2/(K_iV_2 + L_2)]\}} = 0, \quad i = 1, \dots, 3 \quad (n)$$

$$\frac{0.01\{x_{i,4} - [(x_{i,4} + y_{i,3})L_4/(K_iV_3 + L_4)]\}}{\max\{1, x_{i,4} + [(x_{i,4} + y_{i,3})L_4/(K_iV_3 + L_4)]\}} = 0, \quad i = 1, \dots, 3 \quad (o)$$

$$\frac{0.01\{x_{i,6} - [(x_{i,6} + y_{i,5})L_6/(K_iV_5 + L_6)]\}}{\max\{1, x_{i,6} + [(x_{i,6} + y_{i,5})L_6/(K_iV_5 + L_6)]\}} = 0, \quad i = 1, \dots, 3 \quad (p)$$

$$\frac{0.01\{x_{i,8} - [(x_{i,8} + y_{i,7})L_8/(K_iV_7 + L_8)]\}}{\max\{1, x_{i,8} + [(x_{i,8} + y_{i,7})L_8/(K_iV_7 + L_8)]\}} = 0, \quad i = 1, \dots, 3 \quad (q)$$

In summary, the problem consists of 34 bounded variables (both upper bound and lower bounds) associated with the process, 12 linear equality constraints, 18 nonlinear equality constraints, and 3 linear inequality constraints.

4 Solution of the problem. It was not possible to use analytical derivatives in the nonlinear programming code because the energy balance equality constraints

and the process stream phase equilibria constraints involve the stream molar enthalpy H_j and the phase equilibrium constant K_{ij} , respectively. H_j was calculated at zero pressure and then corrected using the Watson acentric factor (Green, 1997). The correction for nonideality was based on correlated experimental data that cannot be differentiated analytically. The component phase equilibrium constant K_{ij} was calculated via the Redlich–Kwong equation of state; the vapor phase mixture compressibility factor z^v was determined as the largest of the three real roots from the virial equation:

$$z^3 - z^2 + C_1 z + C_2 = 0$$

where C_1 and C_2 are functions of the critical properties of the mixture. An analytical derivative of the vapor phase mixture compressibility with respect to the stream variables cannot be determined explicitly, and therefore, the derivative of the component phase equilibrium constant K_{ij} cannot be determined analytically.

As a consequence, the gradient of the objective function and the Jacobian matrix of the constraints in the nonlinear programming problem cannot be determined analytically. Finite difference substitutes as discussed in Section 8.10 had to be used. To be conservative, substitutes for derivatives were computed as suggested by Curtis and Reid (1974). They estimated the ratio μ_j of the truncation error to the roundoff error in the central difference formula

$$\frac{\partial f}{\partial x_j} = \frac{f(x + d_j) - f(x - d_j)}{2d_j}$$

where d_j is the step size, as follows:

$$\mu_j = \frac{-(d_j/2)[f(x + d_j) - 2f(x) + f(x - d_j)]}{d_j^2} \bigg/ p \left| \frac{\partial f}{\partial x_j} \cdot x_j \right|$$

where p is the magnitude of the error incurred in the storage of a number in the computer. The Curtis–Reid method updates d_j on each calculation of a partial derivative from the relation

$$d_j^{k+1} = (d_j^k) \min \left\{ 1000, \sqrt{\frac{\mu_j^*}{\max\{u_j, 1\}}} \right\}$$

where u_j^* is the target value of the error ratio. To ensure that the truncation error calculation was not dominated by round-off error, Curtis and Reid suggested a value for u_j^* of 100 with an acceptable range of 10 to 1000.

The solution listed in Table E15.2B was obtained from several nonfeasible starting points, one of which is shown in Table E15.2C, by the generalized reduced gradient method.

TABLE E15.2B
Final solution of light hydrocarbon refrigeration process

Stream	$F/1000$ $\left(\frac{\text{lb mol}}{\text{h}}\right)$	T (°F)	p (psia)	$H/1000$ $\left(\frac{\text{Btu}}{\text{lb mol}}\right)$	Molar flow rates					
					Liquid/100			Vapor/100		
					C_2H_6	C_3H_8	$n\text{C}_4\text{H}_{10}$	C_2H_6	C_3H_8	$n\text{C}_4\text{H}_{10}$
1	1.00	200	500	6.90	0.00	0.00	0.00	2.50	6.00	1.50
2	2.97	115	500	2.07	10.6	9.15	1.66	5.63	2.46	0.221
3	1.29	80.6	306	4.69	0.00	0.00	0.00	9.04	3.58	0.259
4	1.68	80.6	306	0.0651	7.19	8.03	1.62	0.00	0.00	0.00
5	0.412	33.7	143	4.48	0.00	0.00	0.00	2.86	1.89	0.0749
6	1.27	33.7	143	-1.36	4.34	6.84	1.55	0.00	0.00	0.00
7	0.272	-20.0	511	4.09	0.00	0.00	0.00	1.84	0.843	0.0451
8	1.00	-20.0	511	-2.85	2.50	6.00	1.50	0.00	0.00	0.00
9	1.29	136	500	4.72	0.00	0.00	0.00	9.04	3.58	0.259
10	1.29	50.0	500	-0.373	9.04	3.58	0.259	0.00	0.00	0.00
11	0.412	174	500	6.06	0.00	0.00	0.00	2.86	1.19	0.0749
12	0.412	50.0	500	-0.386	2.86	1.19	0.0749	0.00	0.00	0.00
13	0.272	234	500	7.32	0.00	0.00	0.00	1.84	0.843	0.0451
14	0.272	50.0	500	-0.415	1.84	0.843	0.0451	0.00	0.00	0.00

TABLE E15.2C
Starting point 1 of light hydrocarbon refrigeration optimization

Stream	$F/100$ $\left(\frac{\text{lb mol}}{\text{h}}\right)$	T (°F)	p (psia)	$H/100$ $\left(\frac{\text{Btu}}{\text{lb mol}}\right)$	Molar flow rates					
					Liquid/1000			Vapor/100		
					C_2H_6	C_3H_8	$n\text{C}_4\text{H}_{10}$	C_2H_6	C_3H_8	$n\text{C}_4\text{H}_{10}$
1	1.00	200	500	6.90	0.00	0.00	0.00	2.50	6.00	1.50
2	3.50	103	500	1.79	1.48	0.952	0.167	6.85	1.95	0.149
3	1.47	68.3	300	4.46	0.00	0.00	0.00	1.14	3.12	0.208
4	2.02	68.3	300	-0.156	1.03	0.834	0.161	0.00	0.00	0.00
5	0.372	34.3	175	4.33	0.00	0.00	0.00	2.88	0.793	0.0471
6	1.65	34.3	175	-1.16	0.741	0.755	0.156	0.00	0.00	0.00
7	0.652	-8.17	150	3.46	0.00	0.00	0.00	4.91	1.55	0.0607
8	1.00	-81.7	150	-4.18	0.250	0.600	0.150	0.00	0.00	0.00
9	1.47	125	500	4.70	0.00	0.00	0.00	11.4	3.12	0.208
10	1.47	50.0	500	-0.260	1.14	0.312	0.0208	0.00	0.00	0.00
11	0.372	150	500	5.49	0.00	0.00	0.00	2.88	0.793	0.0471
12	0.372	50.0	500	-0.259	2.88	0.793	0.0471	0.00	0.00	0.00
13	0.652	303	500	8.55	0.00	0.00	0.00	4.91	1.55	0.0607
14	0.652	500	500	-0.290	0.491	0.155	0.0607	0.00	0.00	0.00

15.3 OPTIMIZATION USING MODULAR-BASED SIMULATORS

Over the past 40 years an enormous amount of time and considerable expense have been devoted to the development of modular-based process simulator codes. Figure 15.7 shows typical icons of modules found in a steady-state process simulator, and Figure 15.3 showed the details of one such module. In current practice, optimization meshed with modularly organized simulators prevails because (1) modules are easy to construct and understand, (2) addition and deletion of modules to and from a flowsheet is easily accomplished via a graphical interface without changing the solution strategy, (3) modules are easier to program and debug than sets of equations, and diagnostics for them easier to analyze, and (4) modules already exist and work, whereas equation blocks for equipment have not been prevalent. It seems appropriate, then, to mesh process models in the form of modules with optimization algorithms so that computer codes do not require wholesale rewriting.

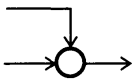
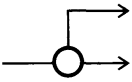

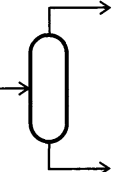
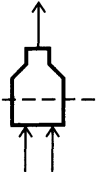
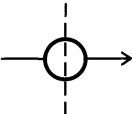
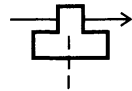
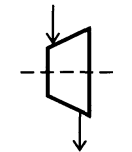
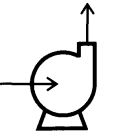
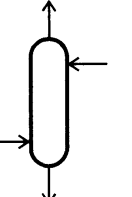
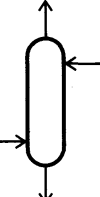
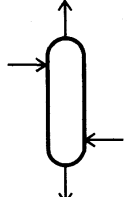
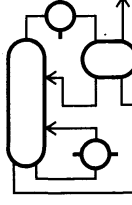
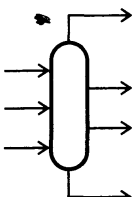
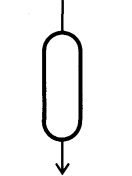
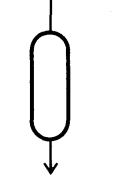

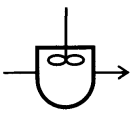
MIXER  Mixer	SPLIT  Splitter	VALVE  Valve	FDRUM  Flash drum	FURNCE  Furnace	EXCHR  Heat exchanger
COMPR  Compressor	TURBN  Turbine	PPUMP  Process pump	ABSOR  Absorber	XTRCT  Extractor	STRIP  Stripper
DISTF  Distillation column	CXCOL  Complex column	RSIMP  Simple reactor	REQUL  Equilibrium reactor	RPLUG  Plug flow reactor	RCSTR  C.S. tank reactor

FIGURE 15.7

Typical process modules used in sequential modular-based flowsheeting codes with their subroutine names.

However, certain difficulties arise in doing this:

1. The input and output variables in a computer module are fixed so that you cannot arbitrarily introduce an output and generate an input, as can be done with an equation-based code.
2. When the modules are connected to one another as represented in a flowsheet, a long train of units may become coupled together for calculations. Thus, a set of modules may require a fixed precedence order of solution so that convergence of the calculations may be slower than in equation-based codes.
3. The modules require some effort to generate reasonably accurate derivatives or their substitutes, especially if a module contains tables, functions with discrete variables, discontinuities, and so on. Perturbation of the input to a module is the primary way in which a finite-difference substitutes for derivatives can be generated.
4. To specify a parameter in a module as a design variable, you need to feed back information around the module and adjust the parameter so that design specifications are met. This arrangement creates a loop exactly the same as a feedback of material or energy creates a recycle loop. Examine Figure 15.8. If the values of many design variables are to be determined, you might end up with several nested loops of calculations (which do, however, enhance stability).
5. Conditions imposed on a process (or a set of equations for that matter) may cause the unit physical states to move from a two-phase to a single-phase operation, or the reverse. As the code shifts from one module to another to represent the process properly, a severe discontinuity occurs in the objective function surface (and perhaps a constraint surface). Derivatives or their substitutes may not change smoothly, and physical property values may jump about.

In Section 15.1 we mentioned that two basic approaches for modular-based process simulators exist:

1. Sequential modular methods.
2. Simultaneous modular methods.

We next consider both methods.

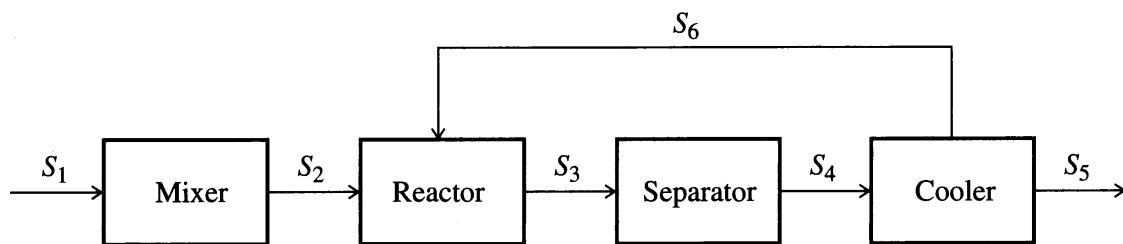


FIGURE 15.8

Modules in which recycling occurs; information (material) from the cooler module is fed back to the reactor, causing a loop.

15.3.1 Sequential Modular Methods

Two procedures are needed to implement efficient computations using sequential calculations in modular-based process simulators: one is *precedence ordering* and the other is *tearing*. Precedence ordering was briefly touched on at the end of Section 15.2 in connection with the partitioning and ordering of equations. The same concept applies to modules connected by loops of information flow. Partitioning the modules in a flowsheet into minimum-size subsets of modules that must be solved simultaneously can be executed by many methods. As with solving sets of equations, to reduce the computational effort you want to obtain the smallest block of modules that constitutes a loop in which the individual modules are tied together by the information flow of outputs and inputs. Between blocks, the information flow occurs serially.

How can you find all of the blocks connected together by information flows? A simple algorithm to isolate blocks is to trace a path of the flow of information (material usually, but possibly energy or a signal) from one module to the next through the module output streams. The tracing continues until either (1) a module in the path is encountered again, in which case all the modules in the path up to the repeated module form a group together that is collapsed and treated as a single module in subsequent tracing, or (2) a module or group with no output is encountered, in which case the module or group of modules can be deleted from the block diagram. As a simple example, examine the block diagram in Figure 15.9, which can be partitioned by the following steps.

Start with an arbitrary unit, say 4, and start tracing the path of information flow in any selected sequence; call this path I:

Start tracing:	4 → 5 → 6 → 4	collapse as one set (456)
Continue tracing:	(456) → 2 → 4	collapse as one set (4562)
Continue tracing:	(4562) → 1 → 2	collapse as one set (45621)
Continue tracing:	(45621) → 7 → 8 → 7	collapse as one set (78)
Continue tracing:	(45621) → (78) → 9	terminate tracing (no output)

The precedence order for path I is as follows:

$$(45621) \rightarrow (78) \rightarrow 9$$

To complete the search and add more modules to the precedence order, start on path II:

Start tracing: 10 → 3 → 2 terminate with 2 as 2 is in path I

The precedence order for path II is

$$(10) \rightarrow (3) \rightarrow (45621)$$

All of the modules from the block diagram have been included in the tracing, and no more paths have to be searched. The procedure identifies all the nested and outer loops. The overall precedence order is (10) → (3) → (45621) → (78) → (9).

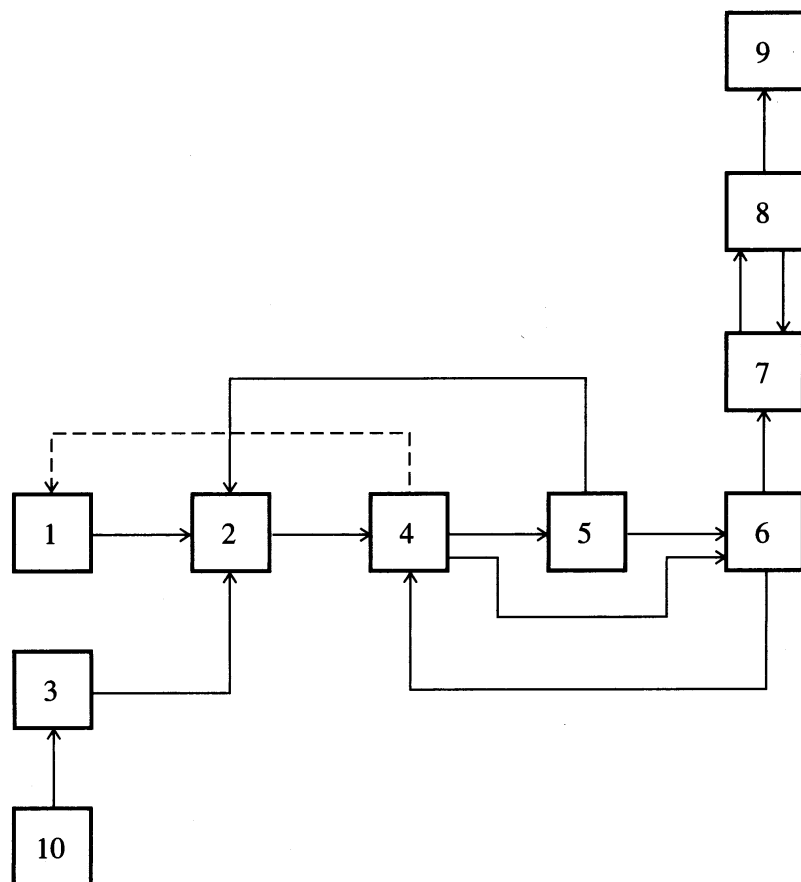


FIGURE 15.9
Block diagram to be partitioned.

Computer techniques to partition complex sets of modules besides the one described earlier can be found in Montagna and Iribarren (1988) and in Mah (1990). Simple sets can be partitioned by inspection.

From a computational viewpoint, the presence of recycle streams is one of the impediments in the sequential solution of a flowsheeting problem. Without recycle streams, the flow of information would proceed in a forward direction, and the calculational sequence for the modules could easily be determined from the precedence order analysis outlined earlier. With recycle streams present, large groups of modules have to be solved simultaneously, defeating the concept of a sequential solution module by module. For example, in Figure 15.8, you cannot make a material balance on the reactor without knowing the information in stream S6, but you have to carry out the computations for the cooler module first to evaluate S6, which in turn depends on the separator module, which in turn depends on the reactor module. Partitioning identifies those collections of modules that have to be solved simultaneously (termed **maximal cyclical subsystems, loops, or irreducible nets**).

To execute a sequential solution for a set of modules, you have to tear certain streams. **Tearing** in connection with modular flowsheeting involves decoupling the interconnections between the modules so that sequential information flow can take place. Tearing is required because of the loops of information created by recycle

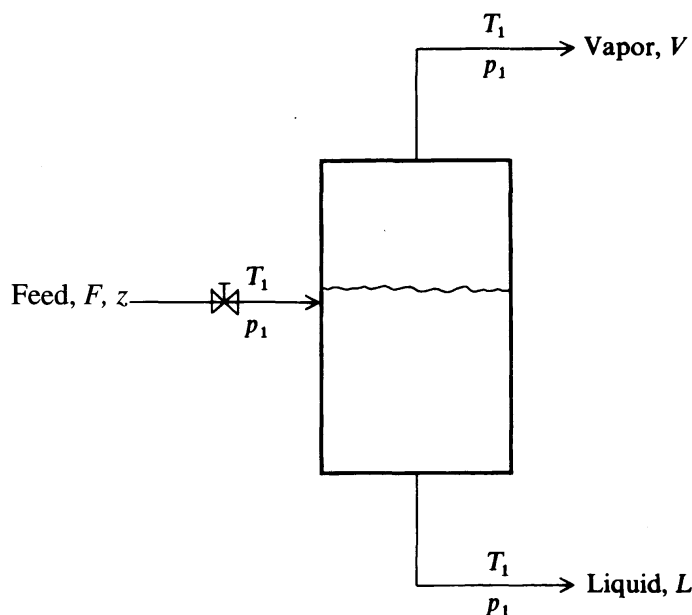


FIGURE 15.10
Vapor–liquid separator.

streams. What you do in tearing is to provide initial guesses for values of some of the unknowns (the tear variables), usually but not necessarily the recycle streams, and then calculate the values of the tear variables from the modules. These calculated values form new guesses, and so on, until the differences between the estimated and calculated values are sufficiently small. **Nesting** of the computations determines which tear streams are to be converged simultaneously and in which order collections of tear or tear streams are to be converged.

Physical insight and experience in numerical analysis are important in selecting which variables to tear. For example, Figure 15.10 illustrates an equilibrium vapor–liquid separator for which the combined material and equilibrium equations give the relation

$$\sum_{j=1}^C \frac{z_j(1 - K_j)}{1 - (V/F) + VK_j/F} = 0$$

where z_j is the mole fraction of species j out of C components in the feed stream, $K_j = y_j/x_j$ is the vapor–liquid equilibrium coefficient, a function of temperature, and the stream flow rates are noted in the figure. For narrow-boiling systems, you can guess V/F , y_j , and x_j , and use the preceding summation to calculate K_j and hence the temperature. This scheme works well because T lies within a narrow range. For wide-boiling materials, the scheme does not converge well. It is better to solve the preceding summation for V/F by guessing T , y , and x , because V/F lies within a narrow range even for large changes in T . Usually, the convergence routines for the code constitute a separator module whose variables are connected to the other modules via the tear variables. Examine Figure 15.11.

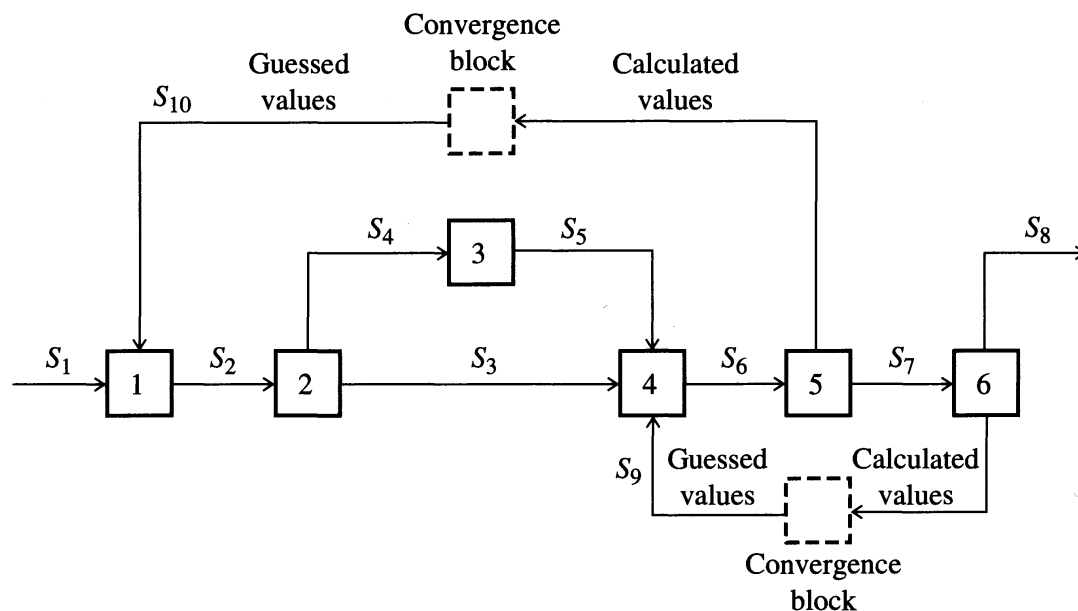


FIGURE 15.11

A computational sequence for modular flowsheeting. Initial values of both recycles are guessed, then the modules are solved in the order 1, 2, 3, 4, 5, and 6. Calculated values for recycle streams S9 and S10 are compared with guessed values in a convergence block, and unless the difference is less than some prescribed tolerance, another iteration takes place with the calculated values, or estimates based on them, forming the new initial guessed values of the recycle streams.

If the objective in selecting streams to tear is to minimize the number of the tear variables (Pho and Lapidus, 1973) subject to the constraint that each loop be broken at least once, this problem is an integer programming problem known as the covering set problem. Refer to Biegler et al. (1997) and Section 8.4.

Although it would logically be quite straightforward to nest the process simulator within the optimization code, and iteratively first satisfy the constraints represented by the simulator by running the simulator, and then applying the optimization code, this procedure is not particularly efficient. The preferred strategy is to insert into the nonlinear optimization problem format, Figure 15.4, the equations corresponding to the convergence blocks in Figure 15.11, namely

$$\tilde{\mathbf{h}}(\mathbf{x}, \mathbf{p}) = \mathbf{0} = \mathbf{x}_T^{(k+1)} - \tilde{\mathbf{f}}(\mathbf{x}, \mathbf{x}_D, \mathbf{x}_T^{(k)}, \mathbf{p})$$

where $\tilde{\mathbf{h}}$ = set of equations involving the tear variables.

$\tilde{\mathbf{f}}$ = set of functions that compute the values of the tear variables for the next iteration ($k + 1$) as the output of a module using the values of tear variables from the previous iteration k .

\mathbf{x}_T = vector of tear variables.

\mathbf{p} = equipment parameter vector

in lieu of using the convergence blocks in the process simulator to determine the values of the tear variables. This procedure saves many iterations through nested loops in the process simulator.

With the preceding implementation, the optimization problem can be solved either via a GRG algorithm or SQP algorithm. Each evaluation of the constraints and objective function requires a full pass through the process simulator. Additional passes are needed to develop the gradients with respect to \mathbf{x} (including the tear variables). Then, the search direction can be obtained as indicated in Figure 15.4 by solving the following QP subproblem.

$$\begin{aligned} \text{Minimize: } & \nabla^T f(\mathbf{x}_I, \mathbf{x}_D, \mathbf{x}_T, \mathbf{p})\mathbf{s} + \frac{1}{2} \mathbf{s}^T \mathbf{B}\mathbf{s} \\ & \mathbf{s} \\ \text{Subject to: } & \mathbf{h}(\mathbf{x}_I, \mathbf{x}_D, \mathbf{x}_T, \mathbf{p}) + \nabla^T \mathbf{h}(\mathbf{x}_I, \mathbf{x}_D, \mathbf{x}_T, \mathbf{p})\mathbf{s} = \mathbf{0} \\ & \tilde{\mathbf{h}}(\mathbf{x}_I, \mathbf{x}_D, \mathbf{x}_T, \mathbf{p}) + \nabla^T \tilde{\mathbf{h}}(\mathbf{x}_I, \mathbf{x}_D, \mathbf{x}_T, \mathbf{p})\mathbf{s} = \mathbf{0} \\ & \mathbf{g}(\mathbf{x}_I, \mathbf{x}_D, \mathbf{x}_T, \mathbf{x}_S, \mathbf{p}) + \nabla^T \mathbf{g}(\mathbf{x}_I, \mathbf{x}_D, \mathbf{x}_T, \mathbf{x}_S, \mathbf{p})\mathbf{s} \geq \mathbf{0} \end{aligned}$$

After determining the search direction \mathbf{s} , an approximate line search is carried out to get the values of \mathbf{x}_I and \mathbf{x}_T for the next iteration.

Of the various versions of the SQP algorithm, the infeasible path reduced SQP has been the most widely used in commercial process simulators. One technique favored by programmers (Lang and Biegler, 1987) is to make just one pass through the process flowsheet simulator before adjusting the values of the decision and tear variables rather than spending considerable computation time satisfying the constraints involved in loops. This procedure has some merit because the value of the variables determined by a fairly precise solution of the loops on one iteration of the optimization program will probably no longer be satisfactory on a subsequent iteration.

15.3.2 Simultaneous Modular Methods

One of the earlier approaches to emulating equation-based optimization using process simulators was to develop by least squares polynomial functions (quadratic being the simplest) to approximate the input–output relations for a module, and for the phase relations (Mahelec et al., 1979; Biegler, 1985; Chen and Stadtherr, 1984; Parker and Hughes (1981); Schmid and Biegler, 1994a). Then, the equations could be used as constraints in an optimization code. Some disadvantages of such an approximation strategy are that (1) adequate approximation of the module may not be possible with simple relations, and (2) the optimum of the approximate model may not lie near the optimum of the rigorous model as ascertained via a more rigorous solution. Nevertheless, such modeling schemes avoid some of the difficulties encountered in closure and convergence of the recycle loops each time the process simulator is called. You obtain the speed and flexibility of the equation-based mode while using as models equations representing the modules.

Use of the reduced space SQP mentioned in Section 15.1 has facilitated the implementation of simultaneous modular optimization. The modeling equations representing the individual modules are not explicitly made part of the optimization problem. Instead, the equations are solved by taking successive steps using Newton's

TABLE 15.2
Comparison of the results of equation-based
and simultaneous modular-based optimization
for two connected distillation columns

Number of variables:	
Decision	4
Outputs, inputs, and so on	47
Internal	114
Number of equality constraints:	
Simultaneous modular strategy*	47
Equation-based strategy	161
Number of iterations (CPU time in seconds)	
Simultaneous modular, SQP	4 (3.4)
Equation-based, SQP	4 (3.3)

Abbreviations: CPU = central processing unit; SQP = successive quadratic programming.

*Method of Schmid and Biegler (1994b)

method for the individual modules. In addition, as proposed by Schmid and Biegler (1994b), a line search is employed that does not require that the Lagrange multipliers associated with the equality constraints be calculated explicitly, an important saving in the case of code modifications. Derivatives are presumed calculated analytically or by finite-difference methods as described in Section 15.3.3. As an example, Table 15.2 lists the results of Schmid and Biegler for the optimization of a hydrodealkylation process (1994b). Comparison of a simultaneous modular strategy with an equation-oriented strategy indicates that both yield equivalent results.

15.3.3 Calculation of Derivatives

Effective computer codes for the optimization of plants using process simulators require accurate values for first-order partial derivatives. In equation-based codes, getting analytical derivatives is straightforward, but may be complicated and subject to error. Analytic differentiation ameliorates error but yields results that may involve excessive computation time. Finite-difference substitutes for analytical derivatives are simple for the user to implement, but also can involve excessive computation time.

For modular-based process simulators, the determination of derivatives is not so straightforward. One way to get partial derivations of the module function(s) is by perturbation of the inputs of the modules in sequence to calculate finite-difference substitutes for derivatives for the torn variables. To calculate the Jacobian via this strategy, you have to simulate each module $(C + 2)n_T + n_F + 1$ times in sequence, where C is the number of chemical species, n_T is the number of torn streams, and n_F is the number of residual degrees of freedom. The procedure is as follows. Start with a tear stream. Back up along the calculation loop until an unperturbed independent variable $x_{i,i}$ in a module is encountered. Perturb the independent variable,

and calculate the resulting dependent and tear variables in that module and all downstream modules in the calculation loop. (Dependent variables upstream are not affected.) Evaluate the finite-difference approximations for the gradients of f , \mathbf{g} , \mathbf{h} , and $\tilde{\mathbf{h}}$ with respect to each $\mathbf{x}_{T,i}$ by using a forward-difference formula in which the values of \mathbf{x}_D are those from the perturbed calculations and the values of \mathbf{x}_T , except $\mathbf{x}_{T,i}$ are perturbed values.

One at a time, perturb the elements of the tear variable $\mathbf{x}_{T,i}$. Calculate the dependent variables, and evaluate the tear equations. Calculate the gradients of f , \mathbf{g} , \mathbf{h} , and $\tilde{\mathbf{h}}$ with respect to each $\mathbf{x}_{T,i}$ by a forward difference equation in which the \mathbf{x}_D are the perturbed values and \mathbf{x}_T are the unperturbed values.

Another way to calculate the partial derivatives is possible. Figure 15.12 represents a typical module. If a module is simulated individually rather than in sequence after each unknown input variable is perturbed by a small amount, to calculate the Jacobian matrix, $(C + 2)n_{ci} + n_{di} + 1$ simulations will be required for the i th module, where n_{ci} = number of interconnecting streams to module i and n_{di} = number of unspecified equipment parameters for module i . This method of calculation of the Jacobian matrix is usually referred to as full-block perturbation.

Wolbert et al. in 1991 proposed a method of obtaining accurate analytical first-order partial derivatives for use in modular-based optimization. Wolbert (1994) showed how to implement the method. They represented a module by a set of algebraic equations comprising the mass balances, energy balance, and phase relations:

$$\Phi_k(\mathbf{u}_k, \mathbf{x}_k) = \mathbf{0} \quad (15.1)$$

where $\Phi_k(\mathbf{u}_k, \mathbf{x}_k)$ = set of functions representing the behavior of the k th module, i.e., the model for module k

\mathbf{u}_k = vector of inputs to the k th module

\mathbf{x}_k = vector of outputs from the k th module

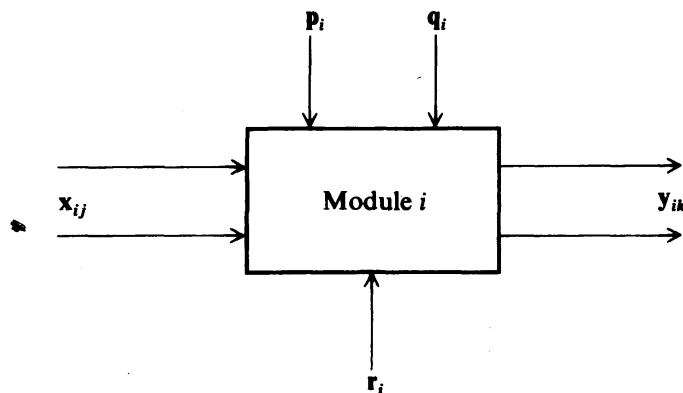


FIGURE 15.12

A typical module showing the input stream vectors \mathbf{x}_{ij} , output stream vectors \mathbf{y}_{ik} , specified equipment parameter vector \mathbf{p}_i , unspecified equipment parameter vector \mathbf{q}_i , and the retention (dependent) variable vector \mathbf{r}_i .

The analytical derivatives $(\partial \phi_{k_i}/\partial x_{k_j})$ and $(\partial \phi_{k_i}/\partial u_{k_j})$, and the sensitivity coefficients $(\partial x_{k_i}/\partial u_{k_j})$ can be obtained directly from Equation (15.1).

As to the automatic generation of exact derivatives in existing modular-based process simulator codes directly from the code itself, refer to Griewank and Corliss (1991) or Bischof et al. (1992).

EXAMPLE 15.3 EXTRACTIVE DISTILLATION DESIGN

This example shows the application of optimization of a process using HYSYS software. Refer to the website, www.mhhe.com/edgar, associated with this book.

EXAMPLE 15.4 MAXIMIZING OPERATING MARGIN

This example shows the application of optimization of a process using Aspen software. Refer to the website, www.mhhe.com/edgar, associated with this book.

15.4 SUMMARY

Commercial process simulators have added optimization capabilities the specific details of which are naturally proprietary, but the general features of these codes are described in this chapter. Very large scale optimization problems of considerable economic value can be treated as shown by the examples presented earlier, and in the future improvements in power, robustness, speed of execution, and user-friendly interfaces of computers and software can be expected to expand the scope of optimization of large scale problems.

REFERENCES

- Aspen Technology, Inc. *Aspen Plus, Aspen Custom Modeler, Dynaplus, Split, Advent, Adsim*. Cambridge, MA (1998).
- Biegler, L. T. "Improved Infeasible Path Optimization for Sequential Modular Simulators—I: The Interface." *Comput Chem Eng* **9**: 245–256 (1985).
- Biegler, L. T.; I. E. Grossmann; and A. W. Westerburg. *Systematic Methods of Chemical Process Design*. Prentice-Hall, Upper Saddle River, NJ (1997).
- Bischof, C.; A. Carle; G. Corliss; A. Griewank; et al. *ADIFOR Generating Derivative Codes for Fortran Programs*. Preprint MCS-P263-0991, Argonne National Lab. (1992).
- ChemCAD. Chemstations. Houston, TX (1998).
- Chen, H. S.; and M. A. Stadtherr. "A Simultaneous-Modular Approach to Process Flow-sheeting and Optimization: I. Theory and Implementation." *AIChE J* **30**: 1843–1856 (1984).

- Curtis, A. R.; and J. K. Reid. "The Choice of Step Lengths When Using Differences to Approximate Jacobian Matrices." *J Inst Math Its Appl* **13**: 121–140 (1974).
- Duff, I. S.; A. M. Erisman; and J. K. Reid. *Direct Methods for Sparse Matrices*. Oxford Univ. Press, New York (1989).
- Fiacco, A. V. "Sensitivity Analysis of Nonlinear Programming Using Penalty Function Methods." *Math Program* **10**: 287–311 (1976).
- Gill, P. E.; W. Murray; and M. H. Wright. *Practical Optimization*. Academic Press, New York (1981).
- Green, D. W., ed. *Perry's Chemical Engineering Handbook*. Section 4, 7th edition. McGraw-Hill, New York (1997).
- Griewank, A.; and G. F. Corliss, eds. *Automatic Differentiation of Algorithms: Theory, Implementation and Application*. SIAM, Philadelphia (1991).
- Gunderson, T.; and I. E. Grossmann. "Improved Optimization Strategies for Automated Heat Exchanger Network Synthesis Through Physical Insights." *Comput Chem Eng* **14**: 925–944 (1990).
- Himmelblau, D. M., ed. *Decomposition of Large Scale Problems*. North-Holland Publ., Amsterdam (1973).
- Hyprotech Ltd. *HYSYM, HYSYS, HYCON*. Calgary, Alberta (1998).
- Intelligen, Inc. *Documentation for EnviroPro Designer and BatchPro Designer*. Scotch Plains, NJ (1999).
- Lang, Y. D.; and L. T. Biegler. "A Unified Algorithm for Flowsheet Optimization." *Comput Chem Eng* **11**: 143–158 (1987).
- Lowery, R. P.; B. McConville; F. H. Yocum; and S. R. Hendon. *Closed-Loop Real Time Optimization of Two Bisphenol-A Plants*. Paper presented at the National AIChE Meeting, Houston, TX, Mar. 28–Apr. 1, 1993.
- Mah, R. H. S. *Chemical Process Structures and Information Flows*. Butterworths (1990).
- Mahalec, V.; H. Kluzik; and L. B. Evans. *Simultaneous Modular Algorithm for Steady State Flowsheet Simulation and Design*. Paper presented at the 12th European Symposium on Computers in Chemical Engineering. Montreaux, Switzerland (1979).
- Montagna, J. M.; and O. A. Iribarren. "Optimal Computation Sequence in the Simulation of Chemical Plants." *Comput Chem Eng* **12**: 12–14 (1988).
- Parker, A. P.; and R. R. Hughes. "Approximate Programming in Chemical Processes—1." *Comput Chem Eng* **5**: 123–133 (1981).
- Perkins, J. D. "Plantwide Optimization—Opportunities and Challenge." In *Foundations of Computer-aided Process Operations*. J. F. Pekny; G. E. Blau, eds. American Institute of Chemical Engineering. New York (1998), pp. 15–26.
- Pho, T. K.; and L. Lapidus. "An Optimum Tearing Algorithm for Recycle Streams." *AIChE J* **19**: 1170–1181 (1973).
- Ramirez, W. F. *Process Control and Identification*. Academic Press, New York (1994).
- Schmid, C.; and L. T. Biegler. "Quadratic Programming Algorithms for Reduced Hessian SQP." *Comput Chem Eng* **18**: 817–832 (1994a).
- Schmid, C.; and L. T. Biegler. "A Simultaneous Approach for Flowsheet Optimization with Existing Modeling Procedures." *Trans Inst Chem Eng* **72A**: May (1994b).
- Seider, W. D.; J. D. Seader; and D. R. Lewin. *Process Design Principles*. Wiley, New York (1999).
- Simulation Sciences, Inc. *Documentation for ROMEO (Rigorous On-line Modeling with Equation-based Optimization)*. Brea, CA (1999).
- Simulation Sciences, Inc. *PRO/II, Provision, Protiss, Hextran*. Brea, CA (1998).
- Westerberg, A. *Advanced System for Computations in Engineering Design*. Report No. ICES 06-239-98, Institute for Complex Engineered Systems, Carnegie-Mellon University (1998).

- Wolbert, D.; X. Joulia; B. Koehret; and L. T. Biegler. "Flowsheet Optimization and Optimal Sensitivity Analysis Using Analytical Derivatives." *Comput Chem Eng* **18**: 1083–1095 (1994).
- Wolbert, D.; X. Joulia; B. Koehret; and M. Pons. "Analyse de Sensibilitie pour l'Optimisation des Procèdes Chimique." 3 d Congres de Genie des Procèdes, Compiegne, France. *Rec Prog Genie Proc* **5**: 415–420 (1991).
- Zaher, J. J. *Condition Modeling*. Ph.D. Thesis, Carnegie-Mellon University, Pittsburgh PA (1995).

SUPPLEMENTARY REFERENCES

- Alkaya, D.; S. Vasantharajan; and L. T. Biegler. "Generalization of a Tailored Approach for Process Optimization." *Ind Eng Chem Res* **39** (6): 1731–1742 (2000).
- Balakrishna, S.; and L. T. Biegler. "A Unified Approach for the Simultaneous Synthesis of Reaction, Energy and Separation Systems." *Ind Eng Chem Res* **32**: 1372–1382 (1993).
- Chen, H. S.; and M. A. Stadtherr. "A Simultaneous Modular Approach to Process Flowsheeting and Optimization." *AIChE J* **31**: 1843–1856 (1985).
- Diwekar, U. M.; I. E. Grossmann; and E. S. Rubin. "MINLP Process Synthesizer for a Sequential Modular Simulator." *Ind Eng Chem Res* **31**: 313–322 (1992).
- Grossmann, I. E.; and M. M. Daichendt. "New Trends in Optimization-Based Approaches to Process Synthesis." *Comput Chem Eng* **20**: 665–683 (1996).
- Kisala, T. P.; R. A. Trevino-Lozano; J. F. Boston; H. I. Britt; et al. "Sequential Modular and Simultaneous Modular Strategies for Process Flowsheet Optimization." *Comput Chem Eng* **11**: 567–579 (1987).
- Kokossis, A. C.; and C. A. Floudas. "Optimization of Complex Reactor Networks—II. Non-isothermal Operation." *Chem Engr Sci* **49** (7): 1037–1051 (1994).
- Kravanja, Z.; and I. E. Grossmann. "Prosyn: An MINLP Process Synthesizer." *Comput Chem Engr* **14**: 1363–1378 (1990).
- Lang, Y. D.; and L. T. Biegler. "A Unified Algorithm for Flowsheet Optimization." *Comput Chem Eng* **11**: 143–158 (1987).
- Pistikopoulos, E. N.; and I. E. Grossmann. "Optimal Retrofit Design for Improving Process Flexibility in Nonlinear Systems—I. Fixed Degree of Flexibility." *Comput Chem Eng* **13**: 1003–1016 (1989).
- Quesada, I.; and I. E. Grossmann. "Global Optimization of Bilinear Process Networks with Multicomponent Streams." *Comput Chem Eng* **19**: 1219–1242 (1995).
- Raman, R.; and I. E. Grossmann. "Symbolic Integration of Logic in Mixed Integer Linear Programming Techniques for Process Synthesis." *Comput Chem Eng* **17**: 909–928 (1993).
- Turkay, M.; and I. E. Grossmann. "Logic-Based MINLP Algorithms for the Optimal Synthesis of Process Networks." *Comput Chem Eng* **20**: 959–978 (1996).

INTEGRATED PLANNING, SCHEDULING,
AND CONTROL IN THE PROCESS
INDUSTRIES

16.1 Plant Optimization Hierarchy	550
16.2 Planning and Scheduling	553
16.3 Plantwide Management and Optimization	565
16.4 Unit Management and Control	567
16.5 Process Monitoring and Analysis	575
References	579
Supplementary References	581

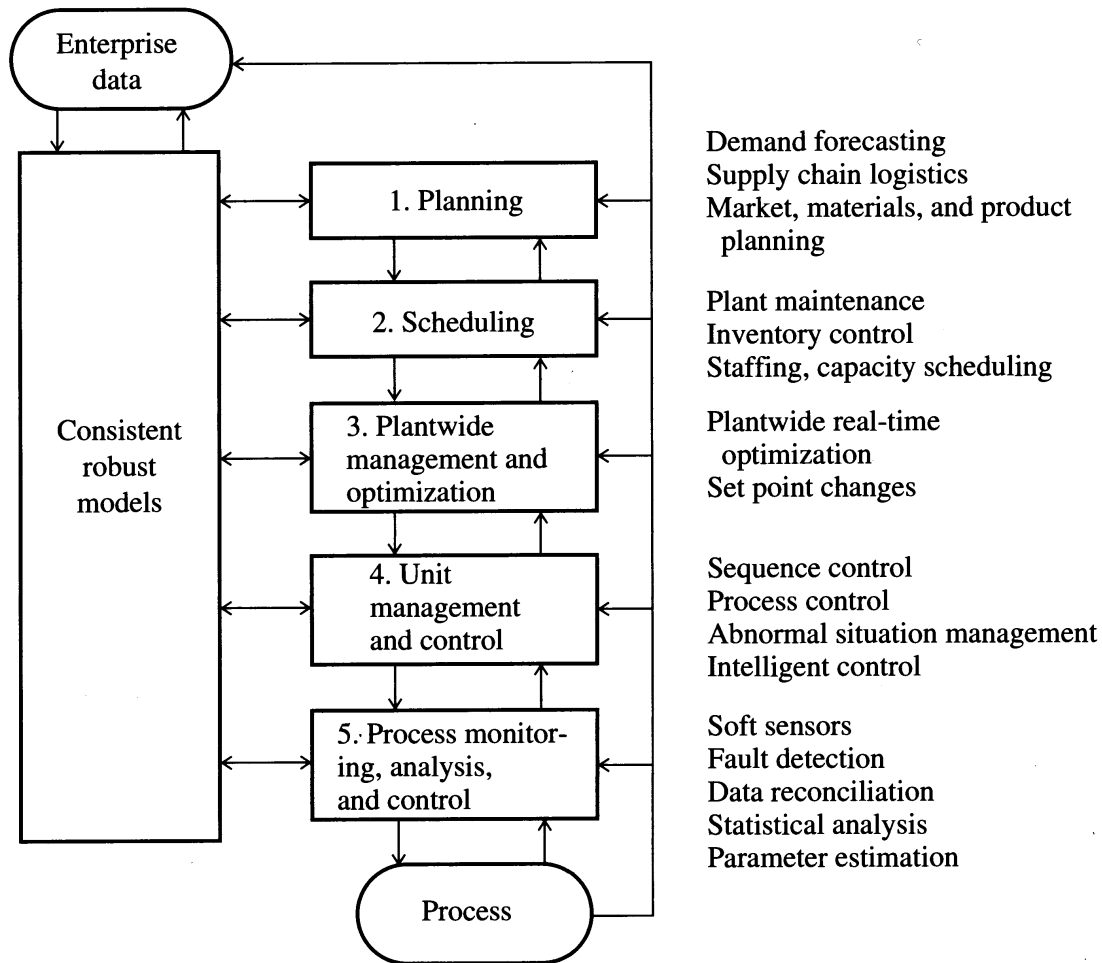
THE COORDINATED USE of computers throughout the entire spectrum of manufacturing and business operations has been growing during the 1990s and is expected to continue during the 21st century. With the continued increases in computing power and advances in telecommunications, the use of optimization has expanded as well, including planning and scheduling, plantwide management, unit management, and data acquisition and monitoring. Coordination of manufacturing with computers has been known since the 1970s as computer-integrated manufacturing (CIM). CIM is defined as a unified network of computer hardware, software, and manufacturing systems that combine business and process functions including administration, economic analysis, scheduling, design, control, operations, interactions among suppliers, multiple plant sites, distribution sites, transportation networks, and customers. Also called *process operations*, the goal of CIM is the management and use of human, capital, material, energy, and information resources to produce desired products safely, flexibly, reliably, and cost-effectively, as rapidly as possible and in an environmentally responsible manner (often characterized as “good, fast, cheap, and clean”).

In the CIM paradigm, operations are guided by extensive interchange of information that integrates sales, marketing, manufacturing, supply, and R&D data. Data and information flow in a seamless fashion among the various sectors. In addition, plant material and energy balance data are analyzed continuously, reconciled using nonlinear programming, and unmeasured variables reconstructed using parameter estimation techniques (soft sensors). General access to a common database and enterprise information are provided to managers, engineers, and operations so that optimum decisions can be made and executed in a timely and efficient manner.

In the remainder of this chapter, we address each part of the manufacturing business hierarchy, and explain how optimization and modeling are key tools that help link the components together.

16.1 PLANT OPTIMIZATION HIERARCHY

Figure 16.1 shows the relevant levels for the process industries in the optimization hierarchy for business manufacturing. At all levels the use of optimization techniques can be pervasive although specific techniques are not explicitly listed in the specific activities shown in the figure. In Figure 16.1 the key information sources for the plant decision hierarchy for operations are the enterprise data, consisting of commercial and financial information, and plant data, usually containing the values of a large number of process variables. The critical linkage between models and optimization in all of the five levels is illustrated in Figure 16.1. The first level (planning) sets production goals that meet supply and logistics constraints, and scheduling (layer 2) addresses time-varying capacity and staffing utilization decisions. The term *supply chain* refers to the links in a web of relationships involving materials acquisition, retailing (sales), distribution, transportation, and manufacturing with suppliers. Planning and scheduling usually take place over relatively long time frames and tend to be loosely coupled to the information flow and analysis that

**FIGURE 16.1**

The five levels of integrated model-based planning, scheduling, optimization, control, and monitoring.

occur at lower levels in the hierarchy. The time scale for decision making at the highest level (planning) may be on the order of months, whereas at the lowest level (e.g., process monitoring) the interaction with the process may be in fractions of a second.

Plantwide management and optimization at level 3 coordinates the network of process units and provides cost-effective setpoints via real-time optimization. The unit management and control level includes process control [e.g., optimal tuning of proportional-integral-derivative (PID) controllers], emergency response, and diagnosis, whereas level 5 (process monitoring and analysis) provides data acquisition and online analysis and reconciliation functions as well as fault detection. Ideally, bidirectional communication occurs between levels, with higher levels setting goals for lower levels and the lower levels communicating constraints and performance information to the higher levels. Data are collected directly at all levels in the enterprise. In practice the decision flow tends to be top down, invariably resulting in mismatches between goals and their realization and the consequent

TABLE 16.1

Types of objective functions and models used in manufacturing system optimization

Optimization level	Objective function	Typical models
1. Planning	Economic	Steady state, single or multiperiod, discrete-event, material flows
2. Scheduling	Economic	Steady state, single or multiperiod, discrete-event, material flows
3. Plantwide management and optimization	Economic	Steady state, linear algebraic correlations or nonlinear simulator
4. Unit management and control		
a. Continuous process	Quadratic-noneconomic or economic	Linear or nonlinear, dynamic, empirical or physically based
b. Batch process	Economic or minimum time	Linear or nonlinear, dynamic or run-to-run, physically based or empirical
5. Process monitoring and analysis		
a. Virtual sensors	Least squares	Nonlinear, physically based, steady state, or empirical
b. Data reconciliation, parameter estimation	Least squares	Linear or nonlinear, steady state or dynamic, physical

accumulation of inventory. Other more deleterious effects include reduction of processing capacity, off-specification products, and failure to meet scheduled deliveries.

Over the past 30 years, business automation systems and plant automation systems have developed along different paths, particularly in the way data are acquired, managed, and stored. Process management and control systems normally use the same databases obtained from various online measurements of the state of the plant. Each level in Figure 16.1 may have its own manually entered database, however, some of which are very large, but web-based data interchange will facilitate standard practices in the future.

Table 16.1 lists the kinds of models and objective functions used in the CIM hierarchy. These models are used to make decisions that reduce product costs, improve product quality, or reduce time to market (or cycle time). Note that models employed can be classified as steady state or dynamic, discrete or continuous, physical or empirical, linear or nonlinear, and with single or multiple periods. The models used at different levels are not normally derived from a single model source, and as a result inconsistencies in the model can arise. The chemical processing industry is, however, moving in the direction of unifying the modeling approaches so that the models employed are consistent and robust, as implied in Figure 16.1. Objective functions can be economically based or noneconomic, such as least

squares. In subsequent sections of this chapter we will demonstrate typical optimization problem formulations for each of the five levels, including decision variables, objective function, and constraints.

16.2 PLANNING AND SCHEDULING

Bryant (1993) states that *planning* is concerned with broad classes of products and the provision of adequate manufacturing capacity. In contrast, *scheduling* focuses on details of material flow, manufacturing, and production, but still may be concerned with offline planning. *Reactive scheduling* refers to real-time scheduling and the handling of unplanned changes in demands or resources. The term *enterprise resource planning* (ERP) is used today, replacing the term manufacturing resources planning (MRP); ERP may or may not explicitly include planning and scheduling, depending on the industry. Planning and scheduling are viewed as distinct levels in the manufacturing hierarchy as shown in Figure 16.1, but often a fair amount of overlap exists in the two problem statements, as discussed later on. The time scale can often be the determining factor in whether a given problem is a planning or scheduling one: planning is typified by a time horizon of months or weeks, whereas scheduling tends to be of shorter duration, that is, weeks, days, or hours, depending on the cycle time from raw materials to final product. Bryant distinguishes among system operations planning, plant operations planning, and plant scheduling, using the tasks listed in Table 16.2. At the systems operations planning level traditional multiperiod, multilocation linear programming problems must be solved, whereas at the plant operations level, nonlinear multiperiod models may be used, with variable time lengths that can be optimized as well (Lasdon and Baker, 1986).

TABLE 16.2
Planning and scheduling hierarchy

Corporate operations planning

- Allocate production requirements to plants.
 - Balance facility's capacity.
 - Optimize materials and product movements (supply chain).
-

Plant operations planning

- Determine production plans.
 - Plan inventory strategy.
 - Determine raw materials requirements.
-

Plant scheduling

- Determine run lengths.
 - Determine sequence of operations.
 - Provide inventory for production runs.
-

Source: Bryant (1993).

Baker (1993) outlined the planning and scheduling activities in a refinery as follows:

1. The corporate operations planning model sets target levels and prices for inter-refinery transfers, crude and product allocations to each refinery, production targets, and inventory targets for the end of each refinery model's time horizon.
2. In plant operations planning each refinery model produces target operating conditions, stream allocations, and blends across the whole refinery, which determines (a) optimal operating conditions, flows, blend recipes, and inventories; and (b) costs, cost limits, and marginal values to the scheduling and real-time optimization (RTO) models.
3. The scheduling models for each refinery convert the preceding information into detailed unit-level directives that provide day-by-day operating conditions or set points.

Supply chain management poses difficult decision-making problems because of its wide ranging temporal and geographical scales, and it calls for greater responsiveness because of changing market factors, customer requirements, and plant availability. Successful supply chain management must anticipate customer requirements, commit to customer orders, procure new materials, allocate production capacity, schedule production, and schedule delivery. According to Bryant (1993), the costs associated with supply chain issues represent about 10 percent of the sales value of domestically delivered products, and as much as 40 percent internationally. Managing the supply chain effectively involves not only the manufacturers, but also their trading partners: customers, suppliers, warehouse, terminal operators, and transportation carriers (air, rail, water, land).

In most supply chains each warehouse is typically controlled according to some local law such as a safety stock level or replenishment rule. This local control can cause buildup of inventory at a specific point in the system and thus propagate disturbances over the time frame of days to months (which is analogous to disturbances in the range of minutes or hours that occur at the production control level). Short-term changes that can upset the system include those that are "self-inflicted" (price changes, promotions, etc.) or effects of weather or other cyclical consumer patterns. Accurate demand forecasting is critical to keeping the supply chain network functioning close to its optimum when the produce-to-inventory approach is used.

16.2.1 Planning

Figure 16.2 shows a simplified and idealized version of the components involved in the planning step, that is, the components of the supply chain. S possible suppliers provide raw materials to each of the M manufacturing plants. These plants manufacture a given product that may be stored or warehoused in W facilities (or may not be stored at all), and these in turn are delivered to C different customers. The nature of the problem depends on whether the products are made to order or made

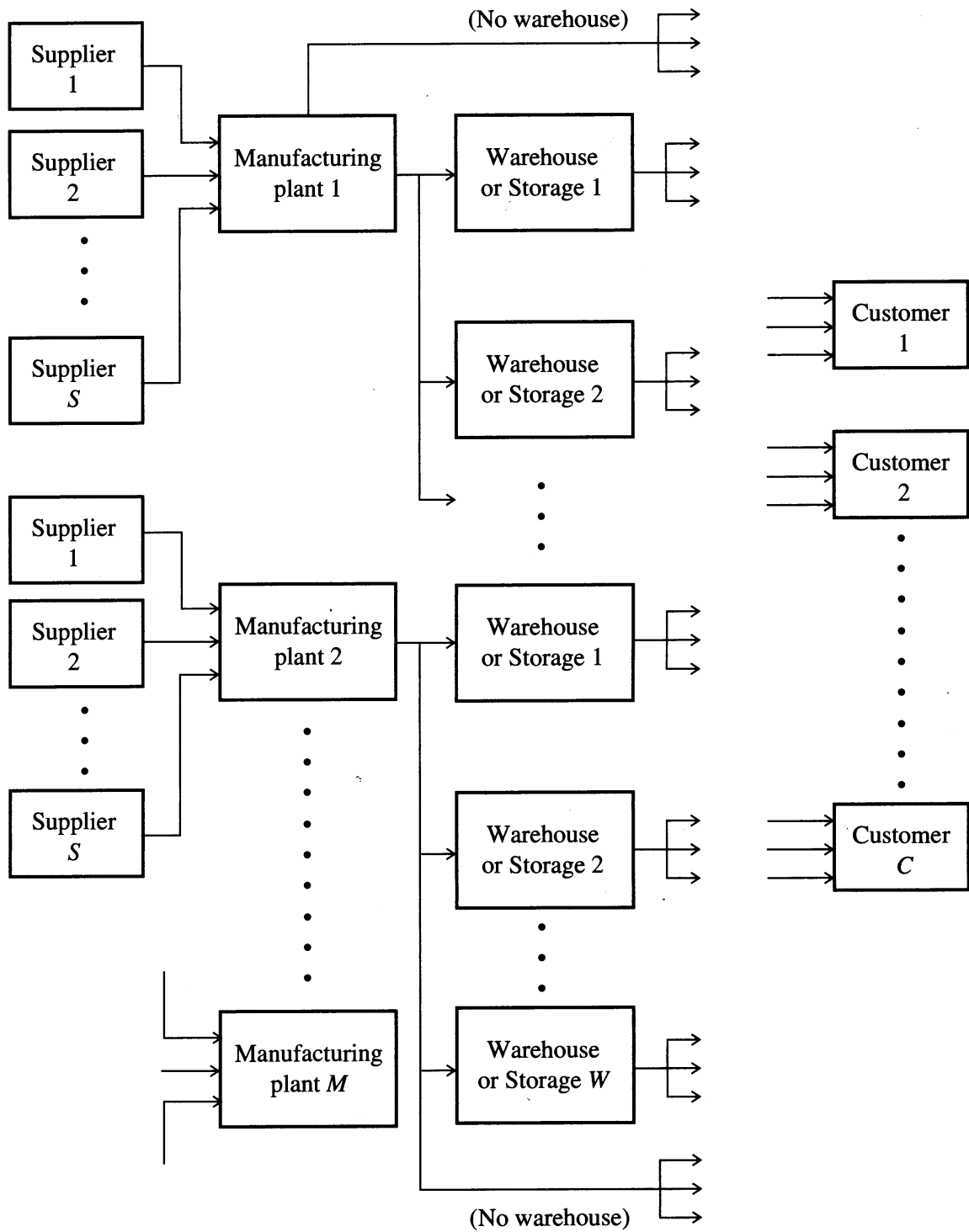


FIGURE 16.2
Supply chain in a manufacturing system.

to inventory; made to order fulfills a specific customer order, whereas made to inventory is oriented to the requirements of the general market demand. Figure 16.2 is similar to a linear allocation process of Chapter 7, with material balance conditions satisfied between suppliers, factories, warehouses, and customers (equality

constraints). Inequality constraints would include individual line capacities in each manufacturing plant, total factory capacity, warehouse storage limits, supplier limits, and customer demand. Cost factors include variable manufacturing costs, cost of warehousing, supplier prices, transportation costs (between each sector), and variable customer pricing, which may be volume and quality-dependent. A practical problem may involve as many as 100,000 variables and can be solved using mixed-integer linear programming (MILP); see Chapter 9.

EXAMPLE 16.1 REFINERY PLANNING AND SCHEDULING

Consider a very simple version of a refinery blending and production problem, which is often formulated and solved in an algebraic modeling language such as GAMS (see Chapters 7 and 9). Figure E16.1 is a schematic of feedstocks and products for the refinery. Table E16.1 lists the information pertaining to the expected yields of the four types of crudes when processed by the refinery. Note that the product distribution from the refinery is quite different for the four crudes. The entire multiunit refinery is aggregated into two processes: a fuel chain and a lube chain. Table E16.1 also lists the forecasted upper limits on the established markets for the various products in terms of the allowed maximum weekly production. The processing costs and other data were taken from Karimi (1992).

The problem is to allocate optimally the crudes between the two processes, subject to the supply and demand constraints, so that profits per week are maximized. The objective function and all constraints are linear, yielding a linear programming problem (LP). To set up the LP you must (1) formulate the objective function and (2) formulate the constraints for the refinery operation. You can see from Figure E16.1 that nine variables are involved, namely, the flow rates of each of the crude oils and the four products.

Solution. We want to decide how much of crudes 1, 2, and 3 should be used in the fuel process, and how much of crude 4 should be allocated to the fuel and the lube processes so as to maximize the weekly profit. One decision variable exists for the amount (kbbbl/wk) of each crude 1, 2, and 3 used in the fuel process. Two variables exist for the amount (kbbbl/wk) of crude 4: one for the amount of crude 4 allocated to the fuel process and the other for the amount allocated to the lube process. Denote the variables by x_c ($c = 1$ to 5), where x_1 through x_3 represent the amounts of crudes 1 through 3, x_4 represents the crude 4 sent to the fuel process, and x_5 represents the crude 4 sent to the lube process. Because the crude supplies are limited, the x_c will be constrained by

$$\begin{aligned}x_1 &\leq S_1 \\x_2 &\leq S_2 \\x_3 &\leq S_3 \\x_4 + x_5 &\leq S_4\end{aligned}\tag{a}$$

where S_c is the maximum supply (kbbbl/wk) of crude c listed in Table E16.1

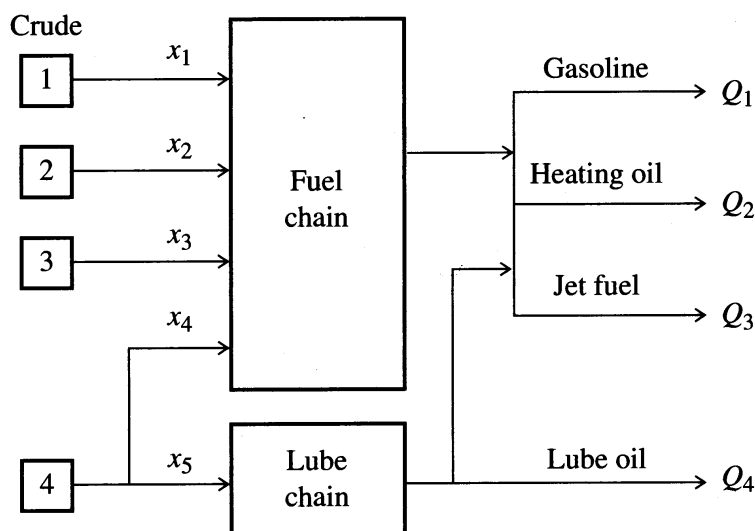


FIGURE E16.1
Processing operation schematic.

TABLE E16.1
Refinery data

	Product yields (bbl/bbl crude)					Product value [selling price (\$/bbl)]	Maximum demand (10^3 bbl/wk)
	Fuel chain				Lube chain		
	Crudes 1 (x_1)	2 (x_2)	3 (x_3)	4 (x_4)	4 (x_5)		
Products							
Gasoline (P_1)	0.6	0.5	0.3	0.4	0.4	45.00	170
Heating oil (P_2)	0.2	0.2	0.3	0.3	0.1	30.00	85
Jet fuel (P_3)	0.1	0.2	0.3	0.2	0.2	15.00	85
Lube oil (P_4)	0.0	0.0	0.0	0.0	0.2	60.00	20
Operating losses	0.1	0.1	0.1	0.1	0.1	—	—
Crude cost (\$/bbl)	15.00	15.00	15.00	25.00	25.00		
Operating cost (\$/bbl)	5.00	8.50	7.50	3.00	2.50		
Available crude supply (10^3 bbl/wk)	100	100	100	200			

Next, we want to find the amounts of different products produced for the given usage x_c of the crudes. Let Q_p ($p = 1$ to 4) refer to the gasoline, heating oil, jet fuel, and lube oil, respectively. Define Q_p as the amount (kbbbl) of product p produced, and let a_{pi} denote the yield of product p from crude x (in bbl/bbl of crude); ($a_{23} = 0.3$, $a_{35} = 0.2$, etc.) Thus, using the a_{pc} from Table E16.1,

$$Q_p = a_{p1}x_1 + a_{p2}x_2 + a_{p3}x_3 + a_{p4}x_4 + a_{p5}x_5, \quad p = 1, \dots, 4 \quad (b)$$

Let D_p be the maximum demand for product p ($D_1 = 170$, etc.). The maximum demands D_p provide the upper bounds on Q_p .

$$Q_p \leq D_p, \quad p = 1, \dots, 4 \quad (c)$$

Finally, we will formulate the objective function. Using the production amounts Q_p and the crude selection x_c , we can calculate the profit as total income from product sales minus the total production cost. If v_p ($p = 1$ to 4) is the value of product p , then total income (k\$) from product sales is $v_1Q_1 + v_2Q_2 + v_3Q_3 + v_4Q_4$. The production cost consists of the costs of crudes and the operating costs. Let C_c ($c = 1$ to 5) denote the sum of crude and operating costs (\$/bbl) for crude usage x_c (e.g., $C_1 = \$20/\text{bbl}$). Then the total production cost is $\sum_{c=1}^5 C_c x_c$. Therefore, the complete problem statement is

$$\text{Maximize:} \quad \sum_{p=1}^4 v_p Q_p - \sum_{c=1}^5 C_c x_c$$

$$\text{Subject to:} \quad x_1 \leq S_1$$

$$x_2 \leq S_2$$

$$x_3 \leq S_3$$

$$x_4 + x_5 \leq S_4$$

$$Q_p \leq D_p \quad (p = 1, \dots, 4) \quad (a)$$

$$Q_p = a_{p1}x_1 + a_{p2}x_2 + a_{p3}x_3 + a_{p4}x_4 + a_{p5}x_5, \quad p = 1, \dots, 4 \quad (b)$$

$$Q_p \geq 0 \quad (p = 1, \dots, 4)$$

$$x_c \geq 0 \quad (c = 1, \dots, 5) \quad (c)$$

The problem involves nine optimization variables (x_c , $c = 1$ to 5 ; Q_p , $p = 1$ to 4) in the preceding formulation. All are continuous variables. The objective function is a linear function of these variables, and so are Equations (a) and (b), hence the problem is a linear programming problem and has a globally optimal solution.

Results. The optimal solution can be obtained using GAMS (Karimi, 1992); the optimum flows are 100, 100, 66.667, and 100 kbbl/wk, respectively, of crudes 1, 2, 3, and 4 and 170, 70, 70, and 20 kbbl/wk, respectively, of gasoline, heating oil, jet fuel, and lube oil are produced. All of crude 4 is used in the lube chain. The maximum profit obtained is 3400 k\$/wk.

As discussed by Karimi (1992), the results for this problem can be interpreted by considering the profit per kilobarrel for each crude. For 1 kbbl/wk of crude 1, we can produce 0.6 kbbl/wk of gasoline, 0.2 kbbl/wk of heating oil and 0.1 kbbl/wk of jet fuel, with production cost of 20 k\$/kbbl/wk and value of the products of $45 * 0.6 + 30 * 0.2 + 15 * 0.1 = 34.5$ k\$. Thus, for 1 kbbl/wk of crude 1, a profit of 14.5 k\$ results. A similar analysis for other crudes yields 8.0 k\$, 4.5 k\$, 2 k\$, and 8.5 k\$, respectively, for crude variables 2, 3, 4, and 5; the priority for the crude options should be 1, 5, 2, 3, and 4. Note that all of crude 1 is used in the optimal solution. Using 100 kbbl/wk of crude 1 produces 60 kbbl/wk of gasoline, 20 kbbl/wk of heating oil, and 10 kbbl/wk of jet fuel. Because this does not exceed the demands of any of the products, the next most

profitable crude (crude 4) can be used in the lube process. Because demand for lube oil cannot be exceeded, only 100 kbbbl/wk of crude 4 can be used in the lube process. Next we can use crude 2, because it does not produce lube oil and is the next most profitable crude. If all of crude 2 (100 kbbbl/wk) is processed, the production amounts become 150, 50, 50, and 20 kbbbl/wk, respectively, but more products can still be manufactured. The maximum amount of crude 3 that can be used without exceeding any of the product demands is 66.667 kbbbl/wk, when the demand of gasoline is equaled. Finally, crude 4 cannot be consumed in the fuel process, because it also produces gasoline and it is not economical to produce any more gasoline.

Most international oil companies that operate multiple refineries analyze the refinery optimization problem over several time periods (e.g., 3 months). This is because many crudes must be purchased at least 3 months in advance due to transportation requirements (e.g., the need to use tankers to transport oil from the Middle East). These crudes also have different grades and properties, which must be factored into the product slate for the refinery. So the multitime period consideration is driven more by supply and demand than by inventory limits (which are typically less than 5 days). The LP models may be run on a weekly basis to handle such items as equipment changes and maintenance, short-term supply issues (and delays in shipments due to weather problems or unloading difficulties), and changes in demand (4 weeks within a 1-month period). Product properties such as the Reid vapor pressure must be changed between summer and winter months to meet environmental restrictions on gasoline properties. See Pike (1986) for a detailed LP refinery example that treats quality specifications and physical properties by using product blending, a dimension not included in Example 16.1 but one that is relevant for companies with varied crude supplies and product requirements.

16.2.2 Scheduling

Information processing in production scheduling is essentially the same as in planning. Both plants and individual process equipment take orders and make products. For a plant, the customer is usually external, but for a process (or “work cell” in discrete manufacturing parlance), the order comes from inside the plant or factory. In a plant, the final product can be sold to an external customer; for a process, the product delivered is an intermediate or partially finished product that goes on to the next stage of processing (internal customer).

Two philosophies are used to solve production scheduling problems (Puigjaner and Espura, 1998):

1. The top-down approach, which defines appropriate hierarchical coordination mechanisms between the different decision levels and decision structures at each level. These structures force constraints on lower operating levels and require heuristic decision rules for each task. Although this approach reduces the size and complexity of scheduling problems, it potentially introduces coordination problems.

TABLE 16.3
Characteristics of batch scheduling
and planning problems

DETERMINE	GIVEN
What Product amounts: lot sizes, batch sizes	Product requirements Horizon, demands, starting and ending inventories
When Timing of specific operations, run lengths	Operational steps Precedence order Resource utilization
Where Sites, units, equipment items	Production facilities Types, capacities
How Resource types and amounts	Resource limitations Types, amounts, rates

Source: Pekny and Reklartitis (1998).

2. The bottom-up approach, which develops detailed plant simulation and optimization models, optimizes them, and translates the results from the simulations and optimization into practical operating heuristics. This approach often leads to large models with many variables and equations that are difficult to solve quickly using rigorous optimization algorithms.

Table 16.3 categorizes the typical problem statement for the manufacturing scheduling and planning problem. In a batch campaign or run, comprising smaller runs called lots, several batches of product may be produced using the same recipe. To optimize the production process, you need to determine

1. The recipe that satisfies product quality requirements.
2. The production rates needed to fulfill the timing requirements, including any precedence constraints.
3. Operating variables for plant equipment that are subject to constraints.
4. Availability of raw material inventories.
5. Availability of product storage.
6. The run schedule.
7. Penalties on completing a production step too soon or too late.

EXAMPLE 16.2 MULTIPRODUCT BATCH PLANT SCHEDULING

Batch operations such as drying, mixing, distillation, and reaction are widely used in producing food, pharmaceuticals, and specialty products (e.g., polymers). Scheduling of operations as described in Table 16.3 is crucial in such plants. A principal feature of batch plants (Ku and Karimi, 1987) is the production of multiple products using the

same set of equipment. Good industrial case studies of plant scheduling include those by Bunch et al. (1998), McDonald (1998), and Schulz et al. (1998). For example, Schulz et al. described a polymer plant that involved four process steps (preparation, reaction, mixing, and finishing) using different equipment in each step. When products are similar in nature, they require the same processing steps and hence pass through the same series of processing units; often the batches are produced sequentially. Such plants are called multiproduct plants. Because of different processing time requirements, the total time required to produce a set of batches (also called the makespan or cycle time) depends on the sequence in which they are produced. To maximize plant productivity, the batches should be produced in a sequence that minimizes the makespan. The plant schedule corresponding to such a sequence can then be represented graphically in the form of a Gantt chart (see the following discussion and Figure E16.2b). The Gantt chart provides a timetable of plant operations showing which products are produced by which units and at what times. Chapter 10 discusses a single-unit sequencing problem.

In this example we consider four products (p_1, p_2, p_3, p_4) that are to be produced as a series of batches in a multiproduct plant consisting of three batch reactors in series (Ku and Karimi, 1992); see Figure E16.2a. The processing times for each batch reactor and each product are given in Table E16.2. Assume that no intermediate storage is available between the processing units. If a product finishes its processing on unit k and unit $k + 1$ is not free because it is still processing a previous product, then the completed product must be kept in unit k , until unit $k + 1$ becomes free. As an example, product p_1 must be held in unit 1 until unit 2 finishes processing p_3 . When a product finishes processing on the last unit, it is sent immediately to product storage. Assume that the times required to transfer products from one unit to another are negligible compared with the processing times.

The problem for this example is to determine the time sequence for producing the four products so as to minimize the makespan. Assume that all the units are initially

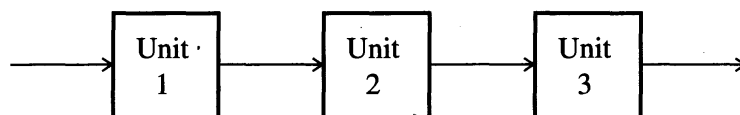


FIGURE E16.2a
Multiproduct plant.

TABLE E16.2
Processing times (h) of products

Units	Products			
	p_1	p_2	p_3	p_4
1	3.5	4.0	3.5	12.0
2	4.3	5.5	7.5	3.5
3	8.7	3.5	6.0	8.0

empty (initialized) at time zero and the manufacture of any product can be delayed an arbitrary amount of time by holding it in the previous unit.

Solution. Let N be the number of products and M be the number of units in the plant. Let $C_{j,k}$ (called completion time) be the “clock” time at which the j th product in the sequence leaves unit k after completion of its processing, and let $\tau_{j,k}$ be the time required to process the j th product in the sequence on unit k (See Table E16.2). The first product goes into unit 1 at time zero, so $C_{1,0} = 0$. The index j in $\tau_{j,k}$ and $C_{j,k}$ denotes the position of a product in the sequence. Hence $C_{N,M}$ is the time at which the last product leaves the last unit and is the makespan to be minimized. Next, we derive the set of constraints (Ku and Karimi, 1988; 1990) that interrelate the $C_{j,k}$. First, the j th product in the sequence cannot leave unit k until it is processed, and in order to be processed on unit k , it must have left unit $k - 1$. Therefore the clock time at which it leaves unit k (i.e., $C_{j,k}$) must be equal to or after the time at which it leaves unit $k - 1$ plus the processing time in k . Thus the first set of constraints in the formulation is

$$C_{j,k} \geq C_{j,k-1} + \tau_{j,k} \quad j = 1, \dots, N \quad k = 2, \dots, M \quad (a)$$

Similarly, the j th product cannot leave unit k until product $(j - 1)$ has been processed and transferred:

$$C_{j,k} \geq C_{j-1,k} + \tau_{j,k} \quad j = 1, \dots, N \quad k = 1, \dots, M \quad (b)$$

Set $C_{0,k} = 0$. Finally the j th product in the sequence cannot leave unit k until the downstream unit $k + 1$ is free [i.e., product $(j - 1)$ has left]. Therefore

$$C_{j,k} \geq C_{j-1,k+1} \quad j = 1, \dots, N \quad k = 1, \dots, M - 1 \quad (c)$$

Although Equations (a)–(c) represent the complete set of constraints, some of them are redundant. From Equation (a) $C_{j,k} \geq C_{j,k-1} + \tau_{i,k}$ for $k \geq 2$. But from Equation (c), $C_{j,k-1} \geq C_{j-1,k}$, hence $C_{j,k} \geq C_{j-1,k} + \tau_{i,k}$ for $k = 2, M$. In essence, Equations (a) and (c) imply Equations (b) for $k = 2, M$, so Equations (b) for $k = 2, M$ are redundant.

Having derived the constraints for completion times, we next determine the sequence of operations. In contrast to the $C_{j,k}$, the decision variables here are discrete (binary). Define $X_{i,j}$ as follows. $X_{i,j} = 1$ if product i (product with label pi) is in slot j of the sequence, otherwise it is zero. So $X_{3,2} = 1$ means that product $p3$ is second in the production sequence, and $X_{3,2} = 0$ means that it is not in the second position. The overall integer constraint is

$$X_{1,j} + X_{2,j} + X_{3,j} + X_{4,j} + \dots + X_{N,j} = 1 \quad j = 1, \dots, \quad (d)$$

Similarly every product should occupy only one slot in the sequence:

$$X_{i,1} + X_{i,2} + X_{i,3} + X_{i,4} + \dots + X_{i,N} = 1 \quad i = 1, \dots, N \quad (e)$$

The $X_{i,j}$ that satisfy Equations (d) and (e) always give a meaningful sequence. Now we must determine the clock times $t_{i,k}$ for any given set of $X_{i,j}$. If product pi is in slot j , then $t_{j,k}$ must be $\tau_{i,k}$ and $X_{i,j} = 1$ and $X_{i,1} = X_{i,2} = \dots = X_{i,j-1} = X_{i,j+1} = \dots = X_{i,N} = 0$, therefore we can use $X_{i,j}$ to pick the right processing time representing $t_{j,k}$ by imposing the constraint.

$$\tau_{j,k} = X_{1,j}t_{1,k} + X_{2,j}t_{2,k} + X_{3,j}t_{3,k} + \dots + X_{N,j}t_{N,k} \quad j = 1, \dots, N \quad k = 1, \dots, M \quad (f)$$

To reduce the number of constraints, we substitute $\tau_{j,k}$ from Equation (f) into Equations (a) and (b) to obtain the following formulation (Ku and Karimi, 1988).

Minimize: C_{NM}

Subject to: Equations (c), (d), (e) and

$$C_{i,k} \geq C_{i,k-1} + \sum_{j=1}^N X_{j,i} t_{j,k} \quad i = 1, \dots, N \quad k = 2, \dots, M \quad (g)$$

$$C_{i,1} \geq C_{i-1,1} + \sum_{j=1}^N X_{j,i} t_{j,i} \quad i = 1, \dots, N \quad (h)$$

$C_{i,k} \geq 0$ and $X_{i,j}$ binary

Because the preceding formulation involves binary ($X_{i,j}$) as well as continuous variables ($C_{i,k}$) and has no nonlinear functions, it is a mixed-integer linear programming (MILP) problem and can be solved using the GAMS MIP solver.

Solving for the optimal sequence using Table E16.2, we obtain $X_{1,1} = X_{2,4} = X_{3,2} = X_{4,3} = 1$. This means that $p1$ is in the first position in the optimal production sequence, $p2$ in the fourth, $p3$ in the second, and $p4$ in the third. In other words, the optimal sequence is in the order $p1$ - $p3$ - $p4$ - $p2$. In contrast to the $X_{i,j}$, we must be careful in interpreting the $C_{i,k}$ from the GAMS output, because $C_{j,k}$ really means the time at which the j th product in the sequence (and not product p_i) leaves unit k . Therefore $C_{2,3} = 23.3$ means that the second product (i.e., $p3$) leaves unit 3 at 23.3 h. Interpreting the others in this way, the schedule corresponding to this production sequence is conveniently displayed in form of a Gantt chart in Figure E16.2b, which shows the status of the units at different times. For instance, unit 1 is processing $p1$ during $[0, 3.5]$ h. When $p1$ leaves unit 1 at $t = 3.5$ h, it starts processing $p3$. It processes $p3$ during $[3.5, 7]$ h. But as seen from the chart, it is unable to discharge $p3$ to unit 2, because unit 2 is still processing $p1$. So unit 1 holds $p3$ during $[7, 7.8]$ h. When unit 2 discharges $p3$

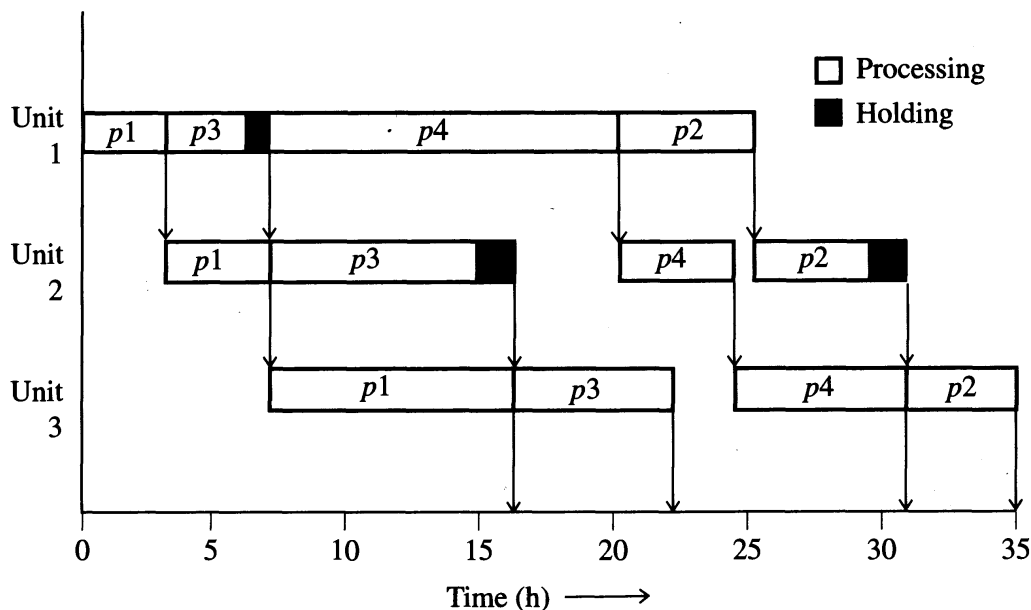


FIGURE E16.2b

Gantt chart for the optimal multiproduct plant schedule.

to unit 3 at 16.5 h, unit 1 is still processing p_4 , therefore unit 2 remains idle during [16.5, 19.8] h. It is common in batch plants to have units blocked due to busy downstream units or units waiting for upstream units to finish. This happens because the processing times vary from unit to unit and from product to product, reducing the time utilization of units in a batch plant. The finished batches of p_1 , p_3 , p_4 , and p_2 are completed at times 16.5 h, 23.3 h, 31.3 h, and 34.8 h. The minimum makespan is 34.8 h.

This problem can also be solved by a search method (see Chapter 10). Because the order of products cannot be changed once they start through the sequence of units, we need only determine the order in which the products are processed. This is the same problem as considered in Section 10.5.2, to illustrate the workings of tabu search. Using the notation of that section, let

$$\mathbf{P} = (p(1), p(2), \dots, p(N))$$

be a permutation or sequence in which to process the jobs, where $p(j)$ is the index of the product in position j of the sequence. To evaluate the makespan of a sequence, we proceed as in Equations (a)–(c) of the mixed-integer programming version of the problem. Let $C_{j,k}$ be the completion time of product $p(j)$ on unit k . If product $p(j)$ does not have to wait for product $p(j-1)$ to finish its processing on unit k , then

$$C_{j,k} = C_{j,k-1} + t_{p(j),k} \quad (i)$$

If it does have to wait, then

$$C_{j,k} = C_{j-1,k} + t_{p(j),k} \quad (j)$$

Hence $C_{j,k}$ is the larger of these two values:

$$C_{j,k} = \max(C_{j-1,k} + t_{p(j),k}, C_{j,k-1} + t_{p(j),k}) \quad (k)$$

This equation is solved first for $C_{1,k}$ for $k = 1, \dots, M$, then for $C_{2,k}$ for $k = 1, 2, \dots, M$, and so on. The objective function is simply the completion time of the last job:

$$f(\mathbf{P}) = C_{N,M} \quad (l)$$

In a four-product problem, there are only $4! = 24$ possible sequences, so you can easily write a simple FORTRAN or C program to evaluate the makespan for an arbitrary sequence, and then call it 24 times and choose the sequence with the smallest makespan. For larger values of N , one can apply the tabu search algorithm described in Section 10.5.2. Other search procedures (e.g., evolutionary algorithms or simulated annealing), can also be developed for this problem. Of course, these algorithms do not guarantee that an optimal solution will be found. On the other hand, the time required to solve the mixed-integer programming formulation grows rapidly with N , so that approach eventually becomes impractical. This illustrates that you may be able to develop a simple but effective search method yourself, and eliminate the need for MILP optimization software.

The classical solution to a scheduling problem assumes that the required information is known at the time the schedule is generated and that this a priori scheduling remains fixed for a planning period and is implemented on the plant equipment. Although this methodology does not compensate for the many external disturbances and internal disruptions that occur in a real plant, it is still the strategy most commonly found in industrial practice. Demand fluctuations, process devia-

tions, and equipment failure all result in schedule infeasibilities that become apparent during the implementation of the schedule. To remedy this situation, frequent rescheduling becomes necessary.

In the rolling horizon rescheduling approach (Baker, 1993), a multiperiod solution is obtained, but only the first period is implemented. After one period has elapsed, we observe the existing inventories, create new demand forecasts, and solve a new multiperiod problem. This procedure tries to compensate for the fixed nature of the planning model. However, as has been pointed out by Pekny and Reklaitis (1998), schedules generated in this fashion generally result in frequent resequencing and reassignment of equipment and resources, which may induce further changes in successive schedules rather than smoothing out the production output. An alternative approach uses a master schedule for planning followed by a reactive scheduling strategy to accommodate changes by readjusting the master schedule in a least cost or least change way.

The terms *able to promise* or *available to promise* (ATP) indicate whether a given customer, product, volume, date, or time request can be met for a potential order. ATP requests might be filled from inventory, unallocated planned production, or spare capacity (assuming additional production). When the production scheduler is content with the current plan, made up of firm orders and forecast orders, the forecast orders are removed but the planned production is left intact. This produces inventory profiles in the model that represent ATP from inventory and from unallocated planned production (Baker, 1993; Smith, 1998).

An important simulation tool used in solving production planning and scheduling problems is the *discrete event dynamic system* (DEDS), which gives a detailed picture of the material flows through the production process. Software for simulating such systems are called discrete event simulators. In many cases, rules or expert systems are used to incorporate the experience of scheduling and planning personnel in lieu of a purely optimization-based approach to scheduling (Bryant, 1993). Expert systems are valuable to assess the effects of changes in suppliers, to locate bottlenecks in the system, and to ascertain when and where to introduce new orders. These expert systems are used in reactive scheduling when fast decisions need to be made, and there is no time to generate another optimized production schedule.

16.3 PLANTWIDE MANAGEMENT AND OPTIMIZATION

At the plantwide management and optimization level (see Figure 16.1), engineers strive for enhancements in the operation of the equipment once it is installed in order to realize the most production, the greatest profit, the minimum cost, the least energy usage, and so on. In plant operations, benefits arise from improved plant performance, such as improved yields of valuable products (or reduced yields of contaminants), better product quality, reduced energy consumption, higher processing rates, and longer times between shut downs. Optimization can also lead to reduced maintenance costs, less equipment wear, and better staff utilization. Optimization can take place plantwide or in combinations of units.

The application of real-time optimization (RTO) in chemical plants has been carried out since the 1960s. Originally a large mainframe computer was used to optimize process setpoints, which were then sent to analog controllers for implementation. In the 1970s this approach, called supervisory control, was incorporated into computer control systems with a distributed microprocessor architecture called a *distributed control system*, or DCS (Seborg et al., 1989). In the DCS both supervisory control and regulatory (feedback) control were implemented using digital computers. Because computer power has increased by a factor of 10^6 over the past 30 years, it is now feasible to solve meaningful optimization problems using advanced tools such as linear or nonlinear programming in real time, meaning faster than the time between setpoint changes.

In RTO (level 3), the setpoints for the process operating conditions are optimized daily, hourly, or even every minute, depending on the time scale of the process and the economic incentives to make changes. Optimization of plant operations determines the setpoints for each unit at the temperatures, pressures, and flow rates that are the best in some sense. For example, the selection of the percentage of excess air in a process heater is quite critical and involves a balance on the fuel–air ratio to ensure complete combustion and at the same time maximize use of the heating potential of the fuel. Examples of periodic optimization in a plant are minimizing steam consumption or cooling water consumption, optimizing the reflux ratio in a distillation column, blending of refinery products to achieve desirable physical properties, or economically allocating raw materials. Many plant maintenance systems have links to plant databases to enable them to track the operating status of the production equipment and to schedule calibration and maintenance. Real-time data from the plant also may be collected by management information systems for various business functions.

The objective function in an economic model in RTO involves the costs of raw materials, values of products, and costs of production as functions of operating conditions, projected sales or interdepartmental transfer prices, and so on.

Both the operating and economic models typically include constraints on

- (a) *Operating Conditions*: Temperatures and pressures must be within certain limits.
- (b) *Feed and Production Rates*: A feed pump has a fixed capacity; sales are limited by market projections.
- (c) *Storage and Warehousing Capacities*: Storage tanks cannot be overfilled during periods of low demand.
- (d) *Product Impurities*: A product may contain no more than the maximum amount of some contaminant or impurity.

In addition, safety or environmental constraints might be added, such as a temperature limit or an upper limit on a toxic species. Several steps are necessary for implementation of RTO, including determining the plant steady-state operating conditions, gathering and validating data, updating of model parameters (if necessary) to match current operations, calculating the new (optimized) setpoints, and implementing these setpoints. An RTO system completes all data transfer, optimization calculations, and setpoint implementations before unit conditions change and require a new optimum to be calculated.

A number of RTO problems characteristic of level 3 in Figure 16.1 have been presented in earlier chapters of this book:

1. Reflux ratio in distillation (Example 12.2).
2. Olefin manufacture (Example 14.1).
3. Ammonia synthesis (Example 14.2).
4. Hydrocarbon refrigeration (Example 15.2).

The last example is particularly noteworthy because it represents the current state of the art in utilizing fundamental process models in RTO.

Another activity in RTO is determining the values of certain empirical parameters in process models from the process data after ensuring that the process is at steady state. Measured variables including flow rates, temperatures, compositions, and pressures can be used to estimate model parameters such as heat transfer coefficients, reaction rate coefficients, catalyst activity, and heat exchanger fouling factors. Usually only a few such parameters are estimated online, and then optimization is carried out using the updated parameters in the model. Marlin and Hrymak (1997) and Forbes et al. (1994) recommend that the updated parameters be observable, represent actual changes in the plant, and significantly influence the location of the optimum; also the optimum of the model should be coincident with that of the true process. One factor in modeling that requires close attention is the accurate representation of the process constraints, because the optimum operating conditions usually lie at the intersection of several constraints. When RTO is combined with model predictive regulatory control (see Section 16.4), then correct (optimal) moves of the manipulated variables can be determined using models with accurate constraints.

Marlin and Hrymak (1997) reviewed a number of industrial applications of RTO, mostly in the petrochemical area. They reported that in practice a maximum change in plant operating variables is allowable with each RTO step. If the computed optimum falls outside these limits, you must implement any changes over several steps, each one using an RTO cycle. Typically, more manipulated variables than controlled variables exist, so some degrees of freedom exist to carry out both economic optimization as well as establish priorities in adjusting manipulated variables while simultaneously carrying out feedback control.

16.4 UNIT MANAGEMENT AND CONTROL

Because of greater integration of plant equipment, tighter quality specifications, and more emphasis on maximum profitability while maintaining safe operating conditions, implementation of advanced multivariable process control is increasing. The distributed control system (DCS) architecture for computer control mentioned in the previous section normally uses feedback control based on a proportional integral derivative (PID) controller at the implementation level for regulatory control (Seborg et al., 1989). Although in principle you can select the three design parameters for PID control in an individual control loop using an optimization technique discussed in Chapter 6 (based on minimizing the sum of squares of the error from

setpoint), this design method is not the normal approach currently taken for level 4 in Figure 16.1 (unit management and control). In industrial practice today, advanced multivariable control strategies are being applied using a mathematical programming approach, which is the main topic of this section.

Model predictive control (MPC) refers to a class of control techniques in which a process model is used to predict the future values of the process outputs, and these predictions are used in computing the best control strategy. The most powerful MPC techniques are based on optimization of a quadratic objective function involving the error between the setpoints and predicted outputs. MPC is especially well suited for difficult multiple-input/multiple-output (MIMO) control problems, in which significant interactions exist between the manipulated inputs and the controlled outputs. In addition, MPC can easily accommodate inequality constraints on the input and output variables, such as upper and lower limits, or rate-of-change limits. The operating goal is to keep the process variables within their limits while moving the process to an economic optimum. The success of model predictive control in solving large multivariable industrial control problems is impressive, perhaps even reaching the status of a “killer” application. Control of units with as many as 60 inputs and 40 outputs is already established in industrial practice. Since the 1970s more than a thousand applications of MPC techniques have been used in oil refineries and petrochemical plants around the world. Thus, MPC has had a substantial influence and is currently the method of choice for difficult multivariable control problems in these industries (Camacho and Bordons, 1999).

A key feature of MPC is that future process behavior is predicted using a dynamic model and the available measurements. The controller outputs are calculated so as to minimize the difference between the predicted process response and the desired response. At each sampling instant the control calculations are repeated and the predictions updated based on current measurements, which is a moving horizon approach. Garcia et al. (1989), Richalet (1993), and Qin and Badgwell (1997) have provided surveys of the MPC approach.

Because empirical dynamic models are generally used, they are only valid over the range of conditions considered during the original plant tests, but MPC can be adapted to optimize plant performance. In this case the control strategy is updated periodically to compensate for changes in process conditions, constraints, or performance criteria. Here the MPC calculations need to be done more frequently (e.g., solving an LP or QP problem at each sampling instant) and thus may require an increased amount of computer resources.

16.4.1 Formulating the MPC Optimization Problem

In MPC a dynamic model is used to predict the future output over the prediction horizon based on a set of control changes. The desired output is generated as a setpoint that may vary as a function of time; the prediction error is the difference between the setpoint trajectory and the model prediction. A model predictive controller is based on minimizing a quadratic objective function over a specific time horizon based on the sum of the square of the prediction errors plus a penalty

related to the square of the changes in the control variable(s). Inequality constraints on the input and output variables can be included in the optimization calculation. At each sampling instant, values of the manipulated variables and controlled variables for the next m time steps are calculated; m is the number of control “moves,” and its selection is discussed later. At each sampling instant, only the first control move (of the m moves that were calculated) is actually implemented. Then, the prediction and control calculations are repeated at the next sampling instant, based on the currently measured state of the process.

In principle, any type of process model can be used to predict future values of the controlled outputs. For example, one can use a physical model based on first principles (e.g., mass and energy balances), a linear model (e.g., transfer function, step response model, or state space-model), or a nonlinear model (e.g., neural nets). Because most industrial applications of MPC have relied on linear dynamic models, later on we derive the MPC equations for a single-input/single-output (SISO) model. The SISO model, however, can be easily generalized to the MIMO models that are used in industrial applications (Lee et al., 1994). One model that can be used in MPC is called the step response model, which relates a single controlled variable y with a single manipulated variable u (based on previous changes in u) as follows:

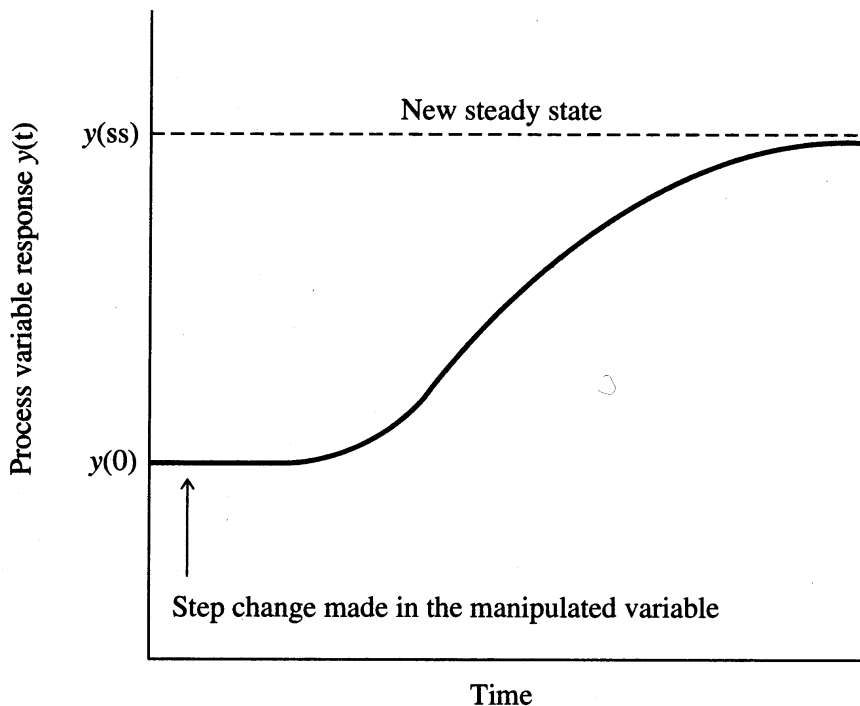
$$\hat{y}(k) = \sum_{i=1}^N S_i \Delta u(k-i) + y(0) \quad (16.1)$$

where $\hat{y}(k)$ is the predicted value of $y(k)$ at the k -sampling instant ($k = 1, 2, \dots$), $\Delta u(k-i)$ is the change in the manipulated input at time $k-i$ [$\Delta u(k-i) = u(k-i) - u(k-i-1)$], N is the number of terms in the step response model (usually less than 50), and the N model parameters S_{ix} are referred to as the step response coefficients. The initial value $y(0)$ is assumed to be known. Other model forms in MPC can involve fewer parameters and can be expressed using state space form (Lee et al., 1994), which is now more frequently used in commercial software packages.

Figure 16.3 shows a hypothetical step response for an industrial process generated by a step change in the manipulated variable u . The model is developed by performing a step change in $u(k)$ and recording the response $y(k)$ until it essentially reaches steady state. In theory, the S_i can be determined from a single-step response but in practice a number of step tests are required to compensate for unanticipated disturbances, process nonlinearities, and noisy measurements. The step response coefficients S_i can be estimated by applying linear regression to the values of the output variable at each sampling instant. Usually the final or steady-state value $y(ss)$ is the last sampled value of y , and the number of data points is selected to be larger than N , the number of terms in the model.

We now develop a mathematical statement for model predictive control with a quadratic objective function for each sampling instant k and linear process model in Equation 16.1:

$$\min f = \sum_{i=1}^p w_i e^2(k+i) + \lambda \sum_{i=1}^m \Delta u^2(k+i-1) \quad (16.2)$$

**FIGURE 16.3**

Typical step response for an industrial process. A time delay may occur between the time that the manipulated variable is changed and the time that the process response occurs.

where $e(k + i)$ denotes the predicted error at time $(k + i)$, $i = 1, \dots, p$,

$$e(k + i) = r(k + i) - \hat{y}(k + i) \quad (16.3)$$

$r(k + i)$ is the reference value or setpoint at time $k + i$, and $\Delta u(k)$ denotes the vector of current and future control moves over the next m sampling instants:

$$\Delta u(k) = [\Delta u(k), \Delta u(k + 1), \dots, \Delta u(k + m - 1)]^T \quad (16.4)$$

To minimize f , you balance the error between the setpoint and the predicted response against the size of the control moves. Equation 16.2 contains design parameters that can be used to tune the controller, that is, you vary the parameters until the desired shape of the response that tracks the setpoint trajectory is achieved (Seborg et al., 1989). The “move suppression” factor λ penalizes large control moves, but the weighting factors w_i allow the predicted errors to be weighted differently at each time step, if desired. Typically you select a value of m (number of control moves) that is smaller than the prediction horizon p , so the control variables are held constant over the remainder of the prediction horizon.

Inequality constraints on future inputs or their rates of change are widely used in the MPC calculations. For example, if both upper and lower limits on u and Δu are required, the constraints could be expressed as

$$B^l \leq u(k + i) \leq B^u, \quad \text{for } i = 1, 2, \dots, m \quad (16.5)$$

$$C^l \leq \Delta u(k + i) \leq C^u, \quad \text{for } i = 1, 2, \dots, m \quad (16.6)$$

where the B^l , C^l , and B^u , C^u are lower and upper bounds, respectively. Note that $u(k + i)$ is determined by whatever values $\Delta u(k + i)$ assume. Constraints on the predicted outputs are sometimes included as well:

$$D^l \leq \hat{y}(k + i) \leq D^u, \quad \text{for } i = 1, 2, \dots, p \quad (16.7)$$

The minimization of the quadratic performance index in Equation (16.2), subject to the constraints in Equations (16.5–16.7) and the step response model such as Equation (16.1), forms a standard quadratic programming (QP) problem, described in Chapter 8. If the quadratic terms in Equation (16.2) are replaced by linear terms, a linear programming program (LP) problem results that can also be solved using standard methods. The MPC formulation for SISO control problems described earlier can easily be extended to MIMO problems and to other types of models and objective functions (Lee et al., 1994). Tuning the controller is carried out by adjusting the following parameters:

- The weighting factor w .
- The move suppression factor λ .
- Bounds for the inputs and input moves.
- The input horizon (m) and output horizon (p).

See the review by Qin and Badgwell (1997) for details on commercial MPC packages.

EXAMPLE 16.3 MODEL PREDICTIVE CONTROL OF A CHEMICAL REACTOR

To carry out changes in the desired operating conditions a chemical reactor is to be controlled using MPC. The reactor is treated as a SISO system; the heat addition rate is the input, and reactor outlet concentration is the output. To design the controller, the system is subjected to a step change in the input, and the output is measured using a constant sampling interval of 1.0 min. Table E16.3 lists the values of the measured output (the response data have been normalized to have a final steady-state value of 1.0). The step response data follow the pattern shown in Figure 16.3. We will use Equation 16.1 to match the step response, with N equal to 70. Once the model coefficients of the response are determined, we can use a QP solver to find the response for a specific setpoint change given the horizons $m = 2$, $p = 4$ for the following three cases:

1. Unconstrained $u(k)$, $\lambda = 0$, $w = 1$
2. $40 \leq u(k) \leq 40$, $\lambda = 0$, $w = 1$
3. Unconstrained $u(k)$; λ is varied using a one-dimensional search (external to the MPC program) to find a good response that satisfies the input constraints in step 2.

Solution. For a given setpoint change you want a smooth, reasonably rapid rise to the new operating point with a small amount of overshoot before settling to the desired

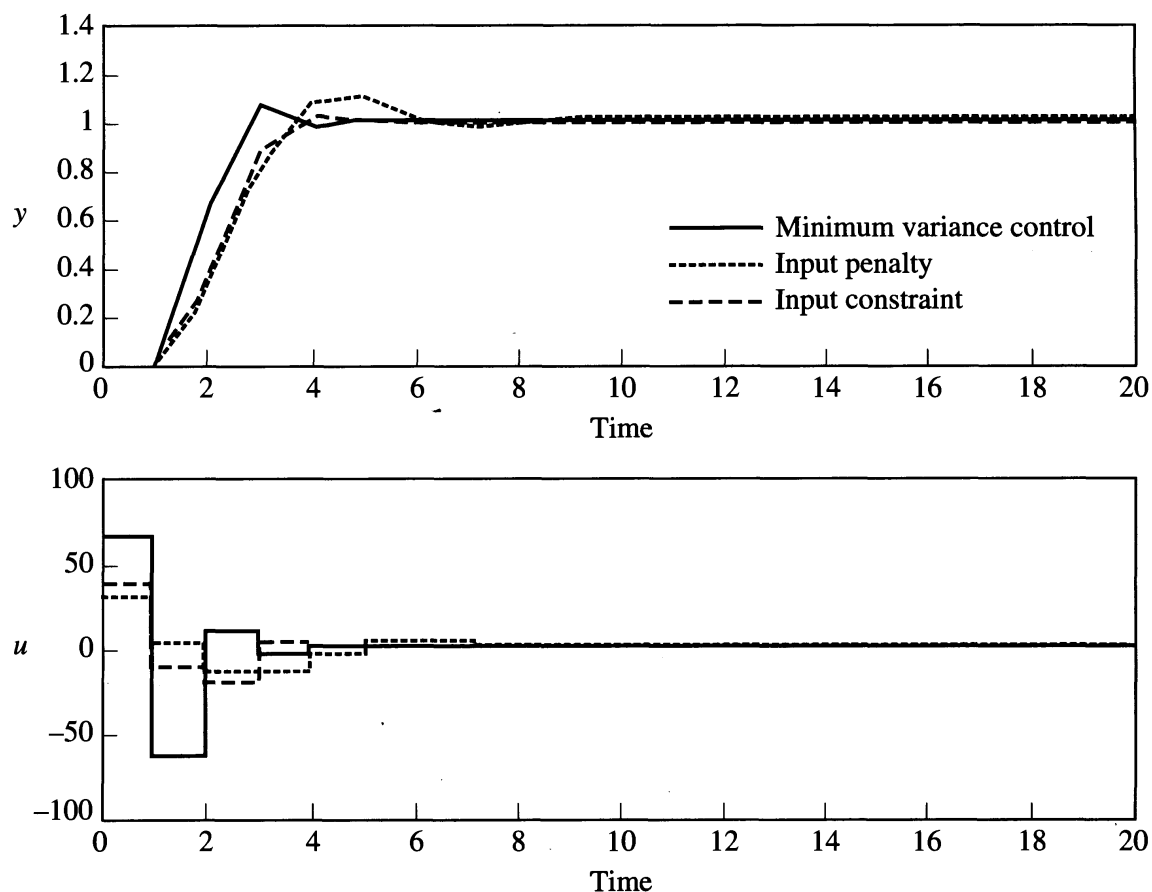
TABLE E16.3
Step response for $\Delta t = 1$

Time	Step response	Time	Step response
1	0.000	36	0.940
2	0.009	37	0.946
3	0.033	38	0.951
4	0.067	39	0.956
5	0.108	40	0.960
6	0.155	41	0.964
7	0.203	42	0.967
8	0.253	43	0.970
9	0.303	44	0.973
10	0.352	45	0.976
11	0.399	46	0.978
12	0.445	47	0.980
13	0.488	48	0.981
14	0.529	49	0.984
15	0.568	50	0.985
16	0.603	51	0.987
17	0.637	52	0.988
18	0.668	53	0.989
19	0.697	54	0.990
20	0.723	55	0.991
21	0.748	56	0.992
22	0.770	57	0.992
23	0.791	58	0.993
24	0.809	59	0.994
25	0.827	60	0.994
26	0.842	61	0.995
27	0.857	62	0.995
28	0.870	63	0.996
29	0.882	64	0.996
30	0.892	65	0.997
31	0.903	66	0.997
32	0.912	67	0.997
33	0.920	68	0.997
34	0.928	69	0.998
35	0.934	70	0.998

operating point. In addition, the changes in the input variable (e.g., valve position for heat transfer medium) should not be too extreme during the transition. Although we do not place a hard limit on the changes in the input, this could easily be done. The step response model for $N = 70$ is simply the values of y for $k = 1$ to 70.

For this example, the controller design was carried out using the MATLAB Model Predictive Control toolbox, which includes a QP solver. Three cases were considered in the preceding problem statement.

1. The MPC controller that minimizes the variance of the output (minimum variance controller) during a setpoint change corresponds to the controller setting $w = 1$, $\lambda = 0$, and no bounds on the input. The response for this controller design for $m = 2$ and $p = 4$ is given in Figure E16.3 by the solid line.

**FIGURE E16.3**

Comparison of the system behavior using three different model predictive controllers (a) minimum variance, (b) input constraint, (c) input penalty.

2. The input for most chemical processes is normally constrained, (e.g., a valve ranges between 0 and 100 percent open). An unconstrained minimum variance controller might not be able to achieve the desired input trajectory for the response. The controller design should take the process input constraints into account. The results of a simulated setpoint change for such a controller with bounds of -40 and 40 for the input and controller parameters $w = 1$ and $\lambda = 0$ is given by the dashed line in Figure E16.3.
3. An alternative method to limit the control action for a controller is to increase the value of the move suppression factor λ , penalizing the change in the input. The system response for small values of λ is close to the unconstrained minimum variance controller as expected, but it violates the constraints. With increasing values of the move suppression factor, however, the second term in Equation (16.2) becomes more important in the objective function, and control changes can correspondingly be limited to the range $-40 \leq u(k) \leq 40$. The dotted line in Figure E16.3 corresponds to a system with the controller setting $w = 1$, $\lambda = 0.01$, and no bounds on the input. Note that the response is much slower than in the direct constraint approach used in case 2.

The control actions in Figure E16.3 are influenced by the choice of the input and output horizon. For this example, all of the controllers had an input horizon of 2 and

an output horizon of 4. In addition to w and λ , the two parameters m and p can be adjusted to improve the response. A selection of shorter horizons will result in more aggressive controllers.

Implementation issues

A critical factor in the successful application of any optimization technique is the availability of a suitable dynamic model. As mentioned previously, in typical MPC applications an empirical model is identified from data acquired during extensive plant tests. The experiments generally consist of a series of step tests, in which the manipulated variables are adjusted one at a time, and the tests require a period of 1–3 weeks. Details concerning the procedures used in the plant tests and subsequent model identification are usually considered to be proprietary information. The scaling and conditioning of plant data for use in model identification and control calculations can be key factors in the success of the application.

Integration of MPC and real-time optimization

Significant potential benefits can be realized by using a combination of MPC and RTO of setpoints that was discussed in Section 16.3. At the present time, most commercial MPC packages integrate the two methodologies in a configuration such as the one shown in Figure 16.4. The MPC calculations are imbedded in the prediction and controller blocks and are carried out quite often (e.g., every 1–10 min). The prediction block predicts the future trajectory of all controlled variables, and the controller achieves the desired response while keeping the process within limits.

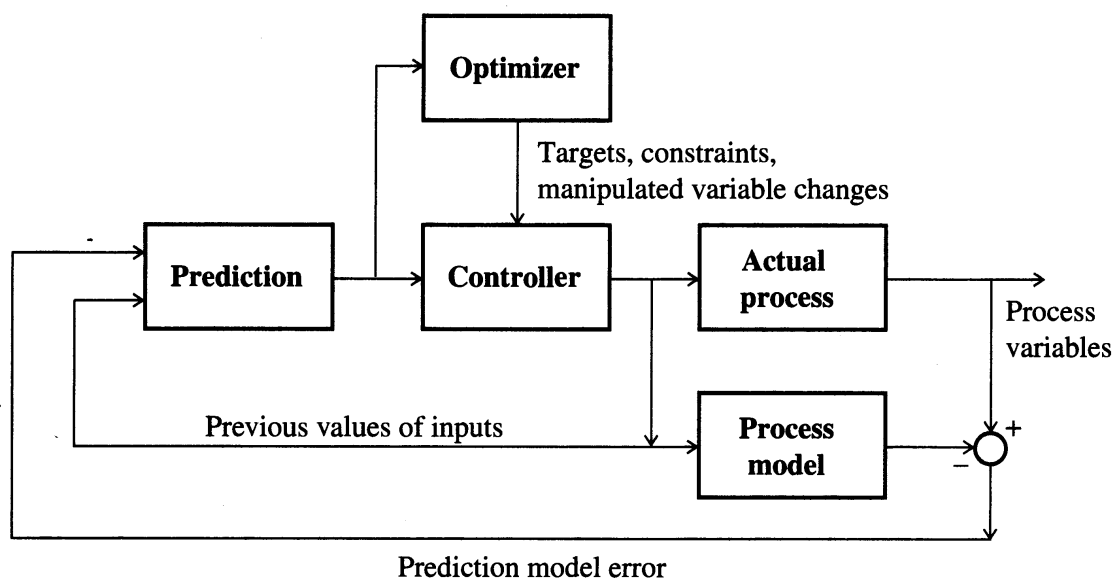


FIGURE 16.4

Diagram showing the combination of real-time optimization and model predictive control in a computer control system.

The targets for the MPC calculations are generated by solving a steady-state optimization problem (LP or QP) based on a linear process model, which also finds the best path to achieve the new targets (Backx et al., 2000). These calculations may be performed as often as the MPC calculations. The targets and constraints for the LP or QP optimization can be generated from a nonlinear process model using a nonlinear optimization technique. If the optimum occurs at a vertex of constraints and the objective function is convex, successive updates of a linearized model will find the same optimum as the nonlinear model. These calculations tend to be performed less frequently (e.g., every 1–24 h) due to the complexity of the calculations and the process models.

16.5 PROCESS MONITORING AND ANALYSIS

Measured process data inherently contain inaccurate information because the measurements are obtained with imperfect instruments. When flawed information is used for estimation of process variables and process control, the state of the system can be misrepresented and the resulting control performance is poor, leading to sub-optimal and even unsafe process operation. Data reconciliation means the adjustment of process data measurements in order to force the data to agree in some sense with a model so that the estimates are better than the data. *Better* is usually defined as the optimal solution to a constrained least squares or maximum likelihood objective function. It is important to understand what is wrong with the values obtained by measurement and why they must be adjusted (Romagnoli and Sanchez, 1999). Data reconciliation can make the process data more useful for decision making and control by smoothing, eliminating outliers, and adjusting for bias and drift, thereby leading to better quality control, detection of faulty instrumentation, detection of process leaks, and increased profits. Computer-integrated manufacturing systems provide plant engineers direct access to extensive plant data as they are recorded. Automation of the data reconciliation computations is necessary to make use of the large amount of information available.

Suppose that the relationship between a measurement of a variable and its true value can be represented by

$$y_m = y + e \quad (16.8)$$

where y_m = measured value

y = true value

e = error

Measurements can contain any of several types of errors: (1) small random errors, (2) systematic biases and drift, or (3) gross errors. Small random errors are zero-mean and are often assumed to be normally distributed (Gaussian). Systematic biases occur when measurement devices provide consistently erroneous values, either high or low. In this case, the expected value of e is not zero. Bias may arise from sources such as incorrect calibration of the measurement device, sensor degradation, damage to the electronics, and so on. The third type of measurement

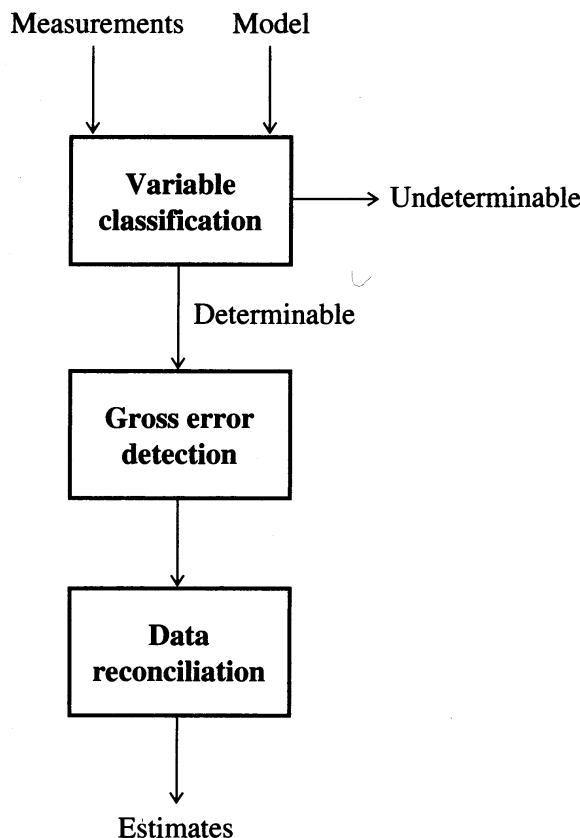


FIGURE 16.5
Steps for data improvement.

error is gross error and is usually caused by large, short-term, nonrandom events. Gross errors can be subdivided into measurement-related errors, such as malfunctioning sensors, and process-related errors, such as process leaks.

Typically, process data are improved using spatial, or functional, redundancies in the process model. Measurements are spatially redundant if more than enough data exist to completely define the process model at any instant, that is, the system is overdetermined and requires a solution by least squares fitting. Similarly, data improvement can be performed using temporal redundancies. Measurements are temporally redundant if past measurement values are available and can be used for estimation purposes. Dynamic models composed of algebraic and differential equations provide both spatial and temporal redundancy.

A simplified view of measurement data improvement techniques can be divided into three basic steps as shown in Figure 16.5. The first step, variable classification, involves determining which variables are observable or unobservable and which are redundant or underdetermined. Several authors have published algorithms for variable classification (Crowe, 1986; Stanley and Mah, 1981; Mah, 1990). Those that are undeterminable are not available for improvement. Next, all gross errors are identified and removed. Several methods proposed for gross error detection have been evaluated by Mah (1990), Rollins et al. (1996) and Tong and Crowe (1997). Data reconciliation concentrates on removing the remaining small, random measurement errors from the data. A key assumption frequently made during the recon-

ciliation step is that the errors are normally distributed, but gross errors severely violate that assumption. If a measurement containing a gross error were allowed into the reconciliation scheme, the resulting estimates of the values of the variable would contain a portion of the gross error distributed among some or perhaps all the estimates (referred to as “smearing”). In practice, gross error detection and elimination are usually performed iteratively along with the final step—data reconciliation.

Historically, treatment of measurement noise has been addressed through two distinct avenues. For steady-state data and processes, Kuehn and Davidson (1961) presented the seminal paper describing the data reconciliation problem based on least squares optimization. For dynamic data and processes, Kalman filtering (Gelb, 1974) has been successfully used to recursively smooth measurement data and estimate parameters. Both techniques were developed for linear systems and weighted least squares objective functions.

The steady-state linear model data reconciliation problem can be stated as

$$\min f = \frac{1}{2}(\hat{y} - y)^T V^{-1}(\hat{y} - y) \quad (16.9)$$

subject to the model constraints

$$A\hat{y} - b = 0 \quad (16.10)$$

where V = variance–covariance matrix (usually diagonal)

y_i = measurement of variable i

\hat{y}_i = reconciled estimate of variable i

A = matrix of linear constraints

b = vector of right-hand side terms in linear constraints

The optimal solution to this problem is

$$\hat{y}^* = [I - VA^T(AVA^T)^{-1}A]y + VA^T(AVA^T)^{-1}b \quad (16.11)$$

If the model includes nonlinear constraints, the problem can be solved using nonlinear programming (Chapter 8).

Several researchers [e.g., Tjoa and Biegler (1992) and Robertson et al. (1996)] have demonstrated advantages of using nonlinear programming (NLP) techniques over such traditional data reconciliation methods as successive linearization for steady-state or dynamic processes. Through the inclusion of variable bounds and a more robust treatment of the nonlinear algebraic constraints, improved reconciliation performance can be realized.

Extended Kalman filtering has been a popular method used in the literature to solve the dynamic data reconciliation problem (Muske and Edgar, 1998). As an alternative, the nonlinear dynamic data reconciliation problem with a weighted least squares objective function can be expressed as a moving horizon problem (Liebman et al., 1992), similar to that used for model predictive control discussed earlier.

The nonlinear objective function (usually quadratic) is

$$\min f(y(t), \hat{y}(t)) \quad (16.12)$$

$$\hat{y}(t)$$

which is subject to the dynamic model

$$h\left(\frac{d\hat{y}(t)}{dt}, \hat{y}(t)\right) = 0 \quad (16.13)$$

and inequality constraints

$$g(\hat{y}(t)) \geq 0 \quad (16.14)$$

This problem can be solved using a combined optimization and constraint model solution strategy (Muske and Edgar, 1998) by converting the differential equations to algebraic constraints using orthogonal collocation or some other model discretization approach.

EXAMPLE 16.4 STEADY-STATE MATERIAL BALANCE RECONCILIATION

Consider the process flowsheet shown in Figure E16.4, which was used by Rollins and Davis (1993) in investigations of gross error detection. The seven stream numbers are identified in Figure E16.4. The overall material balance can be expressed using the constraint matrix $A\hat{y} = 0$, where A is given by

$$A = \begin{bmatrix} 1 & -1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & -1 \end{bmatrix}$$

As a simple case, reconcile a single data set for the stream flows as follows:

$$y = \begin{bmatrix} 49.5 \\ 81.5 \\ 85.3 \\ 10.1 \\ 72.9 \\ 25.7 \\ 50.7 \end{bmatrix}$$

Use the variance-covariance matrix below as a measure of the variability (and reliability) of the stream measurements:

$$V = \begin{bmatrix} 1.5625 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 4.5156 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 4.5156 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.0625 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 3.5156 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.3906 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.3906 \end{bmatrix}$$

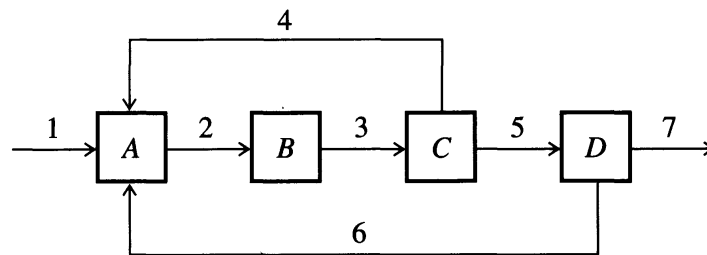


FIGURE E16.4
Recycle process network.

TABLE E16.4
Data reconciliation results

Stream number	True value (kg/min)	Measured value (kg/min)	Reconciled value (kg/min)
1	50.0	49.5	50.0
2	85.0	81.5	85.2
3	85.0	85.3	85.2
4	10.0	10.1	10.0
5	75.0	72.9	75.2
6	25.0	25.7	25.2
7	50.0	50.7	50.0

Solution. The reconciled results in Table E16.4 are obtained by solving the optimization problem with the process model as the only set of constraints. Because all constraints are linear, an analytical solution exists to the problem, as given in Equation 16.11. This results in an 89.6% reduction in the sum of the absolute error. Note that all reconciled values are positive and hence feasible. It is not unusual for some reconciled flow rates to go negative, in which case it is necessary to solve the problem using a constrained minimization code such as QP.

REFERENCES

- Backx, T.; O. Bosgra; and W. Marguardt. "Integration of Model Predictive Control and Optimization of Processes." *ADCHEM Proceedings*, pp. 249–259, Pisa, Italy (2000).
- Baker, T. E. "An Integrated Approach to Planning and Scheduling." In *Foundations of Computer Aided Process Operations (FOCAPO)*, D. W. T. Rippin; J. C. Hale; and J. F. Davis, eds. CACHE Corporation, Austin, TX (1993), pp. 237–252.
- Bryant, G. F. "Developments in Supply Chain Management Control Systems Design." In *Foundations of Computer Aided Process Operations (FOCAPO)*, D. W. T. Rippin; J. C. Hale; and J. F. Davis, eds. CACHE Corporation, Austin, TX (1993), pp. 317–340.

- Bunch, P. R.; D. L. Watson; and J. F. Pekny. "Improving Batch Manufacturing Process Operations Using Mathematical Programming Based Models." *FOCAPO Conference Proceedings, AIChE Symp Ser* 320, **94**: 204–209 (1998).
- Camacho, E. F.; and C. Bordons. *Model Predictive Control*. Springer-Verlag, New York (1999).
- Crowe, C. M. "Reconciliation of Process Flow Rates by Matrix Projection, Part II: The Non-linear Case." *AIChE J* **32**(4): 616–623 (1986).
- Forbes, F.; T. Marlin; and J. MacGregor. "Model Selection Criteria for Economics-Based Optimizing Control." *Comput Chem Eng* **18**: 497–510 (1994).
- Garcia, D. E.; C. E. Prett; and M. Morari. "Model Predictive Control: Theory and Practice—A Survey." *Automatica*, **25**: 335–348 (1989).
- Gelb, A., ed. *Applied Optimal Estimation*. The M.I.T. Press, Cambridge, MA (1974).
- Karimi, I. A. *Refinery Scheduling*. CACHE Process Design Case Studies, I. Grossmann and M. Morari, eds., CACHE Corporation, Austin, TX (1992).
- Ku, H. M.; and I. A. Karimi. "Multiproduct Batch Plant Scheduling." In *CACHE Process Design Case Studies*, I. Grossmann; and M. Morari, eds. CACHE Corporation, Austin, TX (1992).
- Ku, H. M.; and I. A. Karimi. "Completion Time Algorithms for Serial Multiproduct Batch Processes with Shared Storage." *Comput Chem Eng* **14**: 1, 49–59 (1990).
- Ku, H. M.; and I. A. Karimi. "Scheduling in Serial Multiproduct Batch Processes with Finite Interstate Storage: A Mixed Integer Linear Program Formulation." *Ind Eng Chem Res* **27**: 10, 1840 (1988).
- Ku, H. M.; D. Rajagopalan; and I. A. Karimi. "Scheduling in Batch Processes." *Chem Eng Prog* **83**: 8, 35 (1987).
- Kuehn, D. R.; and H. Davidson. "Computer Control." *Chem Eng Prog* **57**(6): 44–47 (1961).
- Lasdon, L. S.; and T. E. Baker. "The Integration of Planning, Scheduling, and Process Control." *Chemical Process Control*, III. T. J. McAvoy; and M. Morari; eds. Elsevier, Amsterdam, Netherlands, (1986), pp. 579–620.
- Lee, J. H.; M. Morari; and C. E. Garcia. "State Space Interpretation of Model Predictive Control." *Automatica* **30**(4): 707–717 (1994).
- Liebman, M. J.; T. F. Edgar; and L. S. Lasdon. "Efficient Data Reconciliation and Estimation for Dynamic Processes Using Nonlinear Programming Techniques." *Comput Chem Eng* **16**(10/11): 963–986 (1992).
- Mah, R. S. H. *Chemical Process Structures and Information Flows*. Butterworth Publishers, Stoneham, MA (1990).
- Marlin, T. E.; and A. N. Hrymak. "Real-Time Operations Optimization of Continuous Processes." In *Chemical Process Control V. AIChE Symp Ser* 316, **93**: 156–164 (1997).
- McDonald, C. M. "Synthesizing Enterprise-Wide Optimization with Global Information Technologies." *FOCAPO Conference Proceedings, AIChE Symp Ser* 320, **94**: 62–74 (1998).
- Muske, K.; and T. F. Edgar. Chapter 6, "Nonlinear State Estimation." In *Nonlinear Process Control*, M. A. Henson; and D. E. Seborg, eds. Prentice-Hall, Englewood Cliffs, NJ (1998).
- Pekny, J. F.; and G. V. Reklaitis. "Towards the Convergence of Theory and Practice: A Technology Guide for Scheduling/Planning Methodology." *FOCAPO Conference Proceedings, AIChE Symp Ser* **94**: 91–111 (1998).
- Pike, R. W. *Optimization for Engineering Systems*. Van Nostrand Reinhold, New York (1986).
- Puigjaner, L.; and A. Espura. "Prospects for Integrated Management and Control of Total Sites in the Batch Manufacturing Industry." *Comput Chem Eng* **22**(1–2): 87–107 (1998).

- Qin, J.; and T. A. Badgwell. "An Overview of Industrial Model Predictive Control Technology." In *Chemical Process Control V, AIChE Symp Ser 316*, **93**: 232–256 (1997).
- Richalet J. "Industrial Applications of Model Based Predictive Control." *Automatica* **29**: 1251–1274 (1993).
- Robertson, D.; J. H. Lee; and J. B. Rawlings. "A Moving Horizon-based Approach for Least Squares State Estimation." *AIChE J* **42**(8): 2209–2224 (1996).
- Rollins, D. K.; and J. F. Davis. "Gross Error Detection When Variance–Covariance Matrices are Unknown." *AIChE J* **39**(8): 1335–1341 (1993).
- Rollins, D. K.; Y. Cheng; and S. Devanathan. "Intelligent Selection of Tests to Enhance Gross Error Identification." *Comput Chem Eng* **20**(5): 517–530 (1996).
- Romagnoli, J. A.; and M. C. Sanchez. *Data Processing and Reconciliation for Chemical Process Operation*. Academic Press, New York (1999).
- Schulz, C.; S. Engell; and R. Rudof. "Scheduling of a Multiproduct Polymer Plant." *FOCAPO Conference Proceedings, AIChE Symp Ser 320*, **94**: 224–230 (1998).
- Seborg, D. E.; T. F. Edgar; and D. A. Mellichamp. *Process Dynamics and Control*. Wiley, New York (1989).
- Smith, W. K. *Time Out*. Wiley, New York (1998).
- Stanley, G. M.; and R. S. H. Mah. "Observability and Redundancy in Process Data." *Chem Eng Sci* **36**: 259–272 (1981).
- Tjoa, I. B.; and L. T. Biegler. "Reduced Successive Quadratic Programming Strategy for Errors-in-Variables Estimation." *Comput Chem Eng* **16**(6): 523–533 (1992).
- Tong, H.; and C. M. Crowe. "Detecting Persistent Gross Errors by Sequential Analysis of Principal Components." *AIChE J* **43**(5): 1242–1249 (1997).

SUPPLEMENTARY REFERENCES

- Boddington, C. E. *Planning, Scheduling, and Control Integration in the Process Industries*. McGraw-Hill, New York (1995).
- Gooding, W. B.; Pekny, J. F.; and P. S. McCroskey. "Enumerative Approaches to Parallel Flowshop Scheduling Via Problem Transformation." *Comput Chem Eng* **18**(10): 909–928 (1994).
- Kondili, E.; Pantelides, C. C.; and R. W. H. Sargent. "A General Algorithm for Short-term Scheduling of Batch Operations—I. MILP Formulation." *Comput Chem Eng* **17**(2): 211–228 (1993).
- Lee J. H.; and B. Cooley. "Recent Advances in Model Predictive Control and Other Related Areas." *Chemical Process Control—V Proceedings, AIChE Symp Ser 316*, **93**: 201–216 (1997).
- Lee, Y. G.; and M. F. Malone. "Batch Process Planning for Waste Minimization." *Ind Eng Chem Res* **39**(6): 2035–2044 (2000).
- Macchieto, S. "Bridging the Gap—Integration of Design, Operations Scheduling, and Control." *FOCAPO-93 Proceedings*, pp. 207–231, CACHE, Austin, TX (1994).
- Morari, M.; J. H. Lee; and C. E. Garcia. *Model Predictive Control*. Prentice-Hall, Englewood Cliffs, NJ (in press).
- Pantelides, C. C. "Unified Frameworks for Optimal Process Planning and Scheduling." In *Foundations of Computer Aided Process Operations*. D. W. T. Rippin; J. C. Hale; and J. F. Davis; eds., pp. 253–274, CACHE, Austin, TX (1994).

- Papageorgaki, S.; and G. V. Reklaitis. "Optimal Design of Multipurpose Batch Plants—I. Problem Formulation." *Ind Eng Chem Res* **29**(10): 2054–2062 (1990).
- Pekny, J. F.; and D. L. Miller. "Exact Solution of the No-Wait Flowshop Scheduling Problem with a Comparison to Heuristic Methods." *Comput Chem Eng* **15**(11): 741–748 (1991).
- Perkins, J. D. "Plant-wide Optimization: Opportunities and Challenges." FOCAPO-98 Proceedings. *AIChE Symp Ser* 320, **94**: 15–26 (1998).
- Shah, N. "Single and Multisite Planning and Scheduling—Current Status and Future Challenges." FOCAPO-98 Proceedings. *AIChE Symp Ser* 320, **94**: 75–90 (1998).
- Shah, N.; C. C. Pantelides; and R. W. H. Sargent. "A General Algorithm for Short-Term Scheduling of Batch Operations. II. Computational Issues." *Comput Chem Eng* **17**(2): 229–244 (1993).
- Shobrys, D. E.; and D. C. White. "Planning, Scheduling, and Control Systems: Why Can They Not Work Together." *Comput Chem Eng* **24**: 163–173 (2000).
- Vision 2020 website: www.chem.purdue.edu/v2020.
- Wright, S. J. "Applying New Optimization Algorithms to Model Predictive Control." Chemical Process Control—V Proceedings. *AIChE Symp Ser* 316, **93**: 147–155 (1994).

APPENDIX A

MATHEMATICAL SUMMARY

A.1	Definitions	584
A.2	Basic Matrix Operations	585
A.3	Linear Independence and Row Operations	593
A.4	Solution of Linear Equations	595
A.5	Eigenvalues, Eigenvectors	598
	References	600
	Supplementary References	600
	Problems	601

THIS APPENDIX SUMMARIZES essential background material concerning matrices and vectors. It is by no means a complete exposition of the subject [see, for example, Stewart (1998), Golub and Van Loan (1996), and Meyer (2000)] but concentrates mainly on those features useful in optimization.

A.1 DEFINITIONS

A matrix is an array of numbers, symbols, functions, and so on

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & a_{22} & \cdots & a_{2m} \\ \vdots & & & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nm} \end{bmatrix} \quad (\text{A.1})$$

An element of the matrix \mathbf{A} is denoted by a_{ij} , where the subscript i corresponds to the row number and subscript j corresponds to the column number. Thus \mathbf{A} in (A.1) has a total of n rows and m columns, and the dimensions of \mathbf{A} are n by m ($n \times m$). If $m = n$, \mathbf{A} is called a “square” matrix. If all elements of \mathbf{A} are zero except the main diagonal (a_{ii} , $i = 1, \dots, n$), \mathbf{A} is called a diagonal matrix. A diagonal matrix with each $a_{ii} = 1$ is called the identity matrix, abbreviated \mathbf{I} .

Vectors are a special type of matrix, defined as having one column and n rows. For example in (A.2) \mathbf{x} has n components

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \quad n \times 1 \text{ matrix, a vector} \quad (\text{A.2})$$

A vector can be thought of as a point in n -dimensional space, although the graphical representation of such a point, when the dimension of the vector is greater than 3, is not feasible. The general rules for matrix addition, subtraction, and multiplication described in Section A.2 apply also to vectors.

The transpose of a matrix or a vector is formed by assembling the elements of the first row of the matrix as the elements of the first column of the transposed matrix, the second row into the second column, and so on. In other words, a_{ij} in the original matrix \mathbf{A} becomes the component a_{ji} in the transpose \mathbf{A}^T . Note that the position of the diagonal components (a_{ii}) are unchanged by transposition. If the dimension of \mathbf{A} is $n \times m$, the dimension of \mathbf{A}^T is $m \times n$ (m rows and n columns). If square matrices \mathbf{A} and \mathbf{A}^T are identical, \mathbf{A} is called a symmetric matrix. The transpose of a vector \mathbf{x} is a row

$$\mathbf{x}^T = [x_1 \quad x_2 \quad \cdots \quad x_n] \quad (\text{A.3})$$

A.2 BASIC MATRIX OPERATIONS

First we present the rules for equality, addition, and multiplication of matrices.

Equality

$$\mathbf{A} = \mathbf{B} \quad \text{if and only if} \quad a_{ij} = b_{ij} \quad \text{for all } i \text{ and } j$$

Furthermore, both \mathbf{A} and \mathbf{B} must have the same dimensions (\mathbf{A} and \mathbf{B} are “conformable”).

Addition

$$\mathbf{A} + \mathbf{B} = \mathbf{C} \quad \text{requires that the element} \quad c_{ij} = a_{ij} + b_{ij}, \quad \text{for all } i \text{ and } j$$

\mathbf{A} , \mathbf{B} , and \mathbf{C} must all have the same dimensions.

Multiplication

$$\mathbf{AB} = \mathbf{C}$$

If the matrix \mathbf{A} has dimensions $n \times m$ and \mathbf{B} has dimensions $q \times r$, then to obtain the product \mathbf{AB} requires that $m = q$ (the number of columns of \mathbf{A} equals the number of rows of \mathbf{B}). The resulting matrix \mathbf{C} is of dimension $n \times r$ and thus depends on the dimensions of both \mathbf{A} and \mathbf{B} . An element c_{ij} of \mathbf{C} is obtained by summing the products of the elements of the i th row of \mathbf{A} times the corresponding elements of the j th column of \mathbf{B} :

$$c_{ij} = \sum_{k=1}^m a_{ik} b_{kj} \quad (\text{A.4})$$

Note that the number of terms in the summation is m , corresponding to the number of columns of \mathbf{A} and the rows of \mathbf{B} . Matrix multiplication in general is not commutative as is the case with scalars, that is,

$$\mathbf{AB} \neq \mathbf{BA}$$

Often the validity of this rule is obvious because the matrix dimensions are not conformable, but even for square matrices commutation is not allowed.

Multiplication of a matrix by a scalar

Each component of the matrix is multiplied by the scalars,

$$s\mathbf{A} = \mathbf{B} \quad \text{is obtained by} \quad s(a_{ij}) = b_{ij} \quad (\text{A.5})$$

Transpose of a product of matrices

The transpose of a matrix product $(\mathbf{AB})^T$ is $(\mathbf{AB})^T = \mathbf{B}^T \mathbf{A}^T$. Likewise, $(\mathbf{ABC})^T = \mathbf{C}^T (\mathbf{AB})^T = \mathbf{C}^T \mathbf{B}^T \mathbf{A}^T$.

EXAMPLE A.1 MATRIX OPERATIONS

Consider a number of simple examples of these operations.

Multiplication:

For

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix}$$

$(3 \times 3) \qquad \qquad (3 \times 2)$

find \mathbf{AB} .

Solution.

$$\mathbf{AB} = \begin{bmatrix} 1(1) + 0(3) + 0(5) & 1(2) + 0(4) + 0(6) \\ 1(1) + 1(3) + 0(5) & 1(2) + 1(4) + 0(6) \\ 1(1) + 1(3) + 1(5) & 1(2) + 1(4) + 1(6) \end{bmatrix} = \begin{bmatrix} 1 & 2 \\ 4 & 6 \\ 9 & 12 \end{bmatrix}$$

(3×2)

Addition:

For

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 2 \\ 1 & -1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} 1 & 5 & 1 \\ 4 & 4 & 4 \\ 1 & 0 & 1 \end{bmatrix}$$

Find $\mathbf{A} + \mathbf{B}$.

Solution.

$$\mathbf{A} + \mathbf{B} = \begin{bmatrix} 1 + 1 & 0 + 5 & 2 + 1 \\ 1 + 4 & -1 + 4 & 0 + 4 \\ 0 + 1 & 0 + 0 & 0 + 1 \end{bmatrix} = \begin{bmatrix} 2 & 5 & 3 \\ 5 & 3 & 4 \\ 1 & 0 & 1 \end{bmatrix}$$

Subtraction:

For

$$\mathbf{A} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} 2 & 6 \\ 1 & 3 \end{bmatrix}$$

find $\mathbf{A} - \mathbf{B}$.

Solution.

$$\mathbf{A} - \mathbf{B} = \begin{bmatrix} 1 - 2 & 1 - 6 \\ 1 - 1 & 1 - 3 \end{bmatrix} = \begin{bmatrix} -1 & -5 \\ 0 & -2 \end{bmatrix}$$

Transpose:

For

$$\mathbf{A} = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} \quad \text{and} \quad \mathbf{B} = \begin{bmatrix} 2 & 1 \\ 3 & 4 \end{bmatrix}$$

find $(\mathbf{AB})^T$.**Solution.**

$$(\mathbf{AB})^T = \mathbf{B}^T \mathbf{A}^T = \begin{bmatrix} 2 & 3 \\ 1 & 4 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} 5 & 2 \\ 5 & 1 \end{bmatrix}$$

Multiplication of matrices by vectors:

A coordinate transformation can be performed by multiplying a matrix times a vector. If

$$\mathbf{A} = \begin{bmatrix} 1 & 1 & 2 \\ 2 & 0 & 3 \\ 4 & 8 & 4 \end{bmatrix} \quad \text{and} \quad \mathbf{x} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

find $\mathbf{y} = \mathbf{Ax}$.

$$\mathbf{y} = \begin{bmatrix} 1(1) + 1(1) + 2(1) \\ 2(1) + 0(1) + 3(1) \\ 4(1) + 8(1) + 4(1) \end{bmatrix} = \begin{bmatrix} 4 \\ 5 \\ 16 \end{bmatrix}$$

Note \mathbf{y} has the same dimension as \mathbf{x} . We have transformed a point in three-dimensional space to another point in that same space.

Other commonly encountered vector–matrix products (\mathbf{x} and \mathbf{y} are n -component vectors) include

$$1. \quad \mathbf{x}^T \mathbf{x} = \sum_{i=1}^n x_i^2 \quad (\text{a scalar}) \quad (\text{A.6})$$

$$2. \quad \mathbf{x}^T \mathbf{y} = \langle \mathbf{x}, \mathbf{y} \rangle = \sum_{i=1}^n x_i y_i \quad (\text{A.7})$$

Equation (A.7) is referred to as the inner product, or dot product, of two vectors. If the two vectors are *orthogonal*, then $\mathbf{x}^T \mathbf{y} = 0$. In two or three dimensions, this means that the vectors \mathbf{x} and \mathbf{y} are perpendicular to each other.

3. $\mathbf{x}^T \mathbf{Ax}$ Here \mathbf{A} is a square matrix of dimension $n \times n$ and the product is a scalar. If \mathbf{A} is a diagonal matrix, then

$$\mathbf{x}^T \mathbf{Ax} = \sum_{i=1}^n a_{ii} x_i^2 \quad (\text{A.8})$$

$$4. \mathbf{xx}^T = \begin{bmatrix} x_1x_1 & x_1x_2 & \cdots & x_1x_n \\ \vdots & \vdots & & \vdots \\ x_nx_1 & & \cdots & x_nx_n \end{bmatrix} \quad (\text{A.9})$$

Each vector has the dimensions $(n \times 1)$ and the matrix is square $(n \times n)$. Note that \mathbf{xx}^T is a matrix rather than a scalar (as with $\mathbf{x}^T\mathbf{x}$).

There is no matrix version of simple division, as with scalar quantities. Rather, the inverse of a matrix (\mathbf{A}^{-1}), which exists only for square matrices, is the closest analog to a divisor. An inverse matrix is defined such that $\mathbf{AA}^{-1} = \mathbf{A}^{-1}\mathbf{A} = \mathbf{I}$ (all three matrices are $n \times n$). In scalar algebra, the equation $a \cdot b = c$ can be solved for b by simply multiplying both sides of the equation by $1/a$. For a matrix equation, the analog of solving

$$\mathbf{AB} = \mathbf{C} \quad (\text{A.10})$$

is to premultiply both sides by \mathbf{A}^{-1} :

$$\begin{aligned} \mathbf{A}^{-1}\mathbf{AB} &= \mathbf{A}^{-1}\mathbf{C} \\ \mathbf{IB} &= \mathbf{A}^{-1}\mathbf{C} \end{aligned} \quad (\text{A.11})$$

Because $\mathbf{IB} = \mathbf{B}$, an explicit solution for \mathbf{B} results. Note that the order of multiplication is critical because of the lack of commutation. Postmultiplication of both sides of Equation (A.10) by \mathbf{A}^{-1} is allowable but does not lead to a solution for \mathbf{B} .

To get the inverse of a diagonal matrix, assemble the inverse of each element on the main diagonal. If

$$\mathbf{A} = \begin{bmatrix} a_{11} & 0 & 0 \\ 0 & a_{22} & 0 \\ 0 & 0 & a_{33} \end{bmatrix}$$

then

$$\mathbf{A}^{-1} = \begin{bmatrix} 1/a_{11} & 0 & 0 \\ 0 & 1/a_{22} & 0 \\ 0 & 0 & 1/a_{33} \end{bmatrix}$$

The proof is evident by multiplication: $\mathbf{AA}^{-1} = \mathbf{I}$.

For a general square matrix of size 2×2 or 3×3 , the procedure is more involved and is discussed later in Examples A.3 and A.7.

The *determinant* (denoted by $\det[\mathbf{A}]$ or $|\mathbf{A}|$) is reasonably easy to calculate by hand for matrices up to size 3×3 :

$$\det \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} = a_{11}a_{22}a_{33} + a_{12}a_{23}a_{31} + a_{13}a_{32}a_{21} - a_{31}a_{22}a_{13} - a_{32}a_{23}a_{11} - a_{33}a_{21}a_{12} \quad (\text{A.12})$$

Another way to calculate the value of a determinant is to evaluate its cofactors. The cofactor of an element a_{ij} of the matrix is found by first deleting from the original matrix the i th row and j th column corresponding to that element; the resulting array is the minor (M_{ij}) for that element and has dimension $(n - 1) \times (n - 1)$. The cofactor is defined as

$$c_{ij} = (-1)^{i+j} \det M_{ij} \quad (\text{A.13})$$

The determinant of the original matrix is calculated by either

$$1. \sum_{j=1}^n a_{ij} c_{ij} \quad (i \text{ fixed arbitrarily; row expansion}) \quad (\text{A.14})$$

or

$$2. \sum_{i=1}^n a_{ij} c_{ij} \quad (j \text{ fixed arbitrarily; column expansion}) \quad (\text{A.15})$$

For example, if

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$$

an expansion of the first row gives

$$\begin{aligned} \det [\mathbf{A}] &= a_{11}c_{11} + a_{12}c_{12} \\ c_{11} &= (-1)^{1+1}a_{22} = a_{22} \\ c_{12} &= (-1)^{1+2}a_{21} = -a_{21} \end{aligned}$$

so that

$$\det [\mathbf{A}] = a_{11}a_{22} - a_{12}a_{21}$$

EXAMPLE A.2 CALCULATE THE VALUE OF A DETERMINANT USING COFACTORS

Calculate the determinant

$$\det \begin{bmatrix} 1 & 2 & 1 \\ 2 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix}$$

using the first row as the expansion.

Solution.

$$\det [\mathbf{A}] = c_{11} + 2c_{12} + c_{13} = \det \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} - 2 \det \begin{bmatrix} 2 & 1 \\ 0 & 1 \end{bmatrix} + \det \begin{bmatrix} 2 & 1 \\ 0 & 0 \end{bmatrix}$$

$$1 = 1 - 4 + 0 = -3$$

It is actually easier to use the third row because of its two zeros.

$$\det \mathbf{A} = c_{33} = (-1)^{3+3} \begin{bmatrix} 1 & 2 \\ 2 & 1 \end{bmatrix} = -3$$

The *adjoint* of a matrix is constructed using the cofactors defined earlier. The elements \bar{a}_{ij} of the adjoint matrix $\bar{\mathbf{A}}$ are defined as

$$\bar{a}_{ij} = c_{ji} \quad (\text{A.16})$$

In other words, the adjoint matrix is the array composed of the transpose of the cofactors.

The adjoint of \mathbf{A} can be used to directly calculate the inverse, \mathbf{A}^{-1} .

$$\mathbf{A}^{-1} = \frac{\text{adj}[\mathbf{A}]}{|\mathbf{A}|} \quad (\text{A.17})$$

Note that the denominator of (A.17), the determinant of $\mathbf{A} \equiv |\mathbf{A}|$, is a scalar. If $|\mathbf{A}| = 0$, the inverse does not exist. A square matrix with determinant equal to zero is called a *singular* matrix. Conversely, for a nonsingular matrix \mathbf{A} , $\det \mathbf{A} \neq 0$.

EXAMPLE A.3 CALCULATION OF THE INVERSE OF A MATRIX

Consider the following matrix and find its inverse.

$$\mathbf{A} = \begin{bmatrix} 1 & 4 \\ 2 & 1 \end{bmatrix} \quad |\mathbf{A}| = 1 - 8 = -7$$

Solution. The cofactors are

$$c_{11} = 1 \quad c_{12} = -2 \quad c_{21} = -4 \quad c_{22} = 1$$

$$\text{adj } \mathbf{A} = \begin{bmatrix} 1 & -4 \\ -2 & 1 \end{bmatrix}$$

$$\mathbf{A}^{-1} = \frac{1}{-7} \begin{bmatrix} 1 & -4 \\ -2 & 1 \end{bmatrix} = \begin{bmatrix} -\frac{1}{7} & \frac{4}{7} \\ \frac{2}{7} & -\frac{1}{7} \end{bmatrix}$$

The use of Equation (A.17) for inversion is conceptually simple, but it is not a very efficient method for calculating the inverse matrix. A method based on use of row operations is discussed in Section A.3. For matrices of size larger than 3×3 , we recommend that you use software such as MATLAB to find \mathbf{A}^{-1} .

Another use for the matrix inverse is to express one set of variables in terms of another, an important operation in constrained optimization (see Chapter 8). For example, suppose \mathbf{x} and \mathbf{z} are two n -vectors that are related by

$$\mathbf{z} = \mathbf{A}\mathbf{x} \quad (\text{A.18})$$

Then, to express \mathbf{x} in terms of \mathbf{z} , merely multiply both sides of (A.18) by \mathbf{A}^{-1} (note that \mathbf{A} must be $n \times n$):

$$\mathbf{A}^{-1}\mathbf{z} = \mathbf{x} \quad (\text{A.19})$$

EXAMPLE A.4 RELATION OF VARIABLES

Suppose that

$$z_1 = x_1 + x_2$$

and

$$z_2 = 2x_1 + x_2$$

What are x_1 and x_2 in terms of z_1 and z_2 ?

Solution. Let

$$\mathbf{z} = \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} \quad \text{and} \quad \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

Therefore $\mathbf{z} = \mathbf{A}\mathbf{x}$, where

$$\mathbf{A} = \begin{bmatrix} 1 & 1 \\ 2 & 1 \end{bmatrix}$$

The inverse of \mathbf{A} is

$$\mathbf{A}^{-1} = \begin{bmatrix} -1 & 1 \\ 2 & -1 \end{bmatrix}$$

hence $\mathbf{x} = \mathbf{A}^{-1}\mathbf{z}$ or

$$x_1 = -z_1 + z_2$$

$$x_2 = 2z_1 - z_2$$

The inverse matrix also can be employed in the solution of linear algebraic equations,

$$\mathbf{A}\mathbf{x} = \mathbf{b} \quad (\text{A.20})$$

which arise in many applications of engineering as well as in optimization theory. To have a unique solution to Equation (A.20), there must be the same number of independent equations as unknown variables. Note that the number of equations is

equal to the number of rows of \mathbf{A} , and the number of unknowns is equal to the number of columns of \mathbf{A} .

With the inverse matrix, you can solve directly for \mathbf{x} :

$$\mathbf{x} = \mathbf{A}^{-1}\mathbf{b} \quad (\text{A.21})$$

Although this is a conceptually convenient way to solve for \mathbf{x} , it is not necessarily the most efficient method for doing so. We shall return to the matter of solving linear equations in Section A.4.

The final matrix characteristic covered here involves differentiation of function of a vector with respect to a vector. Suppose $f(\mathbf{x})$ is a scalar function of n variables (x_1, x_2, \dots, x_n) . The first partial derivative of $f(\mathbf{x})$ with respect to \mathbf{x} is

$$\frac{\partial f}{\partial \mathbf{x}} = \nabla_{\mathbf{x}} f = \left[\frac{\partial f}{\partial x_1} \quad \frac{\partial f}{\partial x_2} \quad \dots \quad \frac{\partial f}{\partial x_n} \right]^T$$

For a vector function $\mathbf{h}(\mathbf{x})$, such as occurs in a series of nonlinear multivariable constraints

$$h_1(x_1, x_2, \dots, x_n) = 0$$

$$h_2(x_1, x_2, \dots, x_n) = 0$$

$$\vdots$$

$$h_m(x_1, x_2, \dots, x_n) = 0$$

the matrix of first partial derivatives, called the Jacobian matrix, is

$$\mathbf{J} = \frac{\partial \mathbf{h}}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\partial h_1}{\partial x_1} & \frac{\partial h_1}{\partial x_2} & \dots & \frac{\partial h_1}{\partial x_n} \\ \vdots & \vdots & & \vdots \\ \frac{\partial h_m}{\partial x_1} & \frac{\partial h_m}{\partial x_2} & \dots & \frac{\partial h_m}{\partial x_n} \end{bmatrix}$$

For a scalar function, the matrix of second derivatives, called the Hessian matrix, is

$$\mathbf{H}(\mathbf{x}) \equiv \nabla^2 f = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \dots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \dots & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\ \vdots & \vdots & & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \dots & \dots & \frac{\partial^2 f}{\partial x_n^2} \end{bmatrix}$$

The use of this matrix and its eigenvalue properties is discussed in several chapters. For continuously differentiable functions, \mathbf{H} is symmetric.

A.3 LINEAR INDEPENDENCE AND ROW OPERATIONS

As mentioned earlier, singular matrices have a determinant of zero value. This outcome occurs when a row or column contains all zeros or when a row (or column) in the matrix is linearly dependent on one or more of the other rows (or columns). It can be shown that for a square matrix, row dependence implies column dependence. By definition the columns of \mathbf{A} , \mathbf{a}_j , are linearly independent if

$$\sum_{j=1}^n d_j \mathbf{a}_j = \mathbf{0} \quad \text{only if } d_j = 0 \text{ for all } j \quad (\text{A.22})$$

Conversely, linear dependence occurs when some nonzero set of values for d_j satisfies Equation (A.22). The *rank* of a matrix is defined as the number of linearly independent columns ($\leq n$).

EXAMPLE A.5 LINEAR INDEPENDENCE AND THE RANK OF A MATRIX

Calculate the rank of

$$\mathbf{A} = \begin{bmatrix} 1 & -1 & 1 \\ 1 & 0 & 1 \\ 2 & -2 & 2 \end{bmatrix}$$

Solution. Note that columns 1 and 3 are identical. Likewise the third row can be formed by multiplying the first row by 2. Equation (A.22) is

$$d_1 \begin{bmatrix} 1 \\ 1 \\ 2 \end{bmatrix} + d_2 \begin{bmatrix} -1 \\ 0 \\ -2 \end{bmatrix} + d_3 \begin{bmatrix} 1 \\ 1 \\ 2 \end{bmatrix} = \mathbf{0}$$

One solution of (A.22) is $d_1 = 1$, $d_2 = 0$, $d_3 = -1$. Because a nontrivial (nonzero) solution exists, then the matrix has one dependent and two independent columns, and the rank ≤ 2 (here 2). The determinant is zero, as can be readily verified using Equation (A.12).

In general for a matrix, the determination of linear independence cannot be performed by inspection. For large matrices, rather than solving the set of linear equations (A.22), elementary row or column operations can be used to demonstrate linear

independence. These operations involve adding some multiple of one row to another row, analogous to the types of algebraic operations (discussed later) that are used to solve simultaneous equations. The value of the determinant of \mathbf{A} is invariant under these row (or column) operations. Implications with respect to linear independence and the use of determinants for equation-solving are discussed in Section A.4.

EXAMPLE A.6 USE OF ROW OPERATIONS

Use row operations to determine if the matrix

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 2 & 2 \\ 3 & 4 & 5 \end{bmatrix}$$

is nonsingular, that is, composed of linearly independent columns.

Solution. First create zeros in the a_{21} and a_{31} position by multiplication or addition. The necessary transformations are

1. Multiply row 1 by (-1) ; add to row 2

$$\mathbf{C}_1 = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 2 & 1 \\ 3 & 4 & 5 \end{bmatrix}$$

2. Multiply row 1 by (-3) ; add to row 3

$$\mathbf{C}_2 = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 2 & 1 \\ 0 & 4 & 2 \end{bmatrix}$$

Next use row 2 to create a zero in a_{32} .

3. Multiply row 2 by (-2) ; add to row 3

$$\mathbf{C}_3 = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 2 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$

Note that neither rows 1 or 2 are changed in this step. The appearance of a row with all zero elements indicates that the matrix is singular ($\det [\mathbf{A}] = 0$).

Row operations can also be used to obtain an inverse matrix. Suppose we augment \mathbf{A} with an identity matrix \mathbf{I} of the same dimension; then multiply the augmented matrix by \mathbf{A}^{-1} :

$$\mathbf{A}^{-1}[\mathbf{A} \mid \mathbf{I}] = [\mathbf{I} \mid \mathbf{A}^{-1}] \quad (\text{A.23})$$

If \mathbf{A} is transformed by row operations to obtain \mathbf{I} , \mathbf{A}^{-1} occurs in the augmented part of the matrix.

EXAMPLE A.7 CALCULATION OF INVERSE MATRIX

Verify the results of Example A.3 using row operations.

Solution. Form the augmented matrix

$$\mathbf{C}_0 = \begin{bmatrix} 1 & 4 & 1 & 0 \\ 2 & 1 & 0 & 1 \end{bmatrix}$$

Successive transformations would be

$$\mathbf{C}_1 = \begin{bmatrix} 1 & 4 & 1 & 0 \\ 0 & -7 & -2 & 1 \end{bmatrix} \quad \mathbf{C}_2 = \begin{bmatrix} 1 & 4 & 1 & 0 \\ 0 & 1 & \frac{2}{7} & -\frac{1}{7} \end{bmatrix}$$

$$\mathbf{C}_3 = \begin{bmatrix} 1 & 0 & -\frac{1}{7} & \frac{4}{7} \\ 0 & 1 & \frac{2}{7} & -\frac{1}{7} \end{bmatrix}$$

Therefore the inverse of \mathbf{A} is

$$\mathbf{A}^{-1} = \begin{bmatrix} -\frac{1}{7} & \frac{4}{7} \\ \frac{2}{7} & -\frac{1}{7} \end{bmatrix}$$

A.4 SOLUTION OF LINEAR EQUATIONS

The need to solve sets of linear equations arises in many optimization applications. Consider Equation (A.20), where \mathbf{A} is an $n \times n$ matrix corresponding to the coefficients in n equations in n unknowns. Because $\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}$, then from (A.17) $|\mathbf{A}|$ must be nonzero; \mathbf{A} must have rank n , that is, no linearly dependent rows or columns exist, for a unique solution. Let us illustrate two cases where $|\mathbf{A}| = 0$:

$$2x_1 + 2x_2 = 6$$

$$x_1 + x_2 = 5$$

or

$$\begin{bmatrix} 2 & 2 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 6 \\ 5 \end{bmatrix}$$

It is obvious that only one linearly independent column or row exists, and $|\mathbf{A}|$ is zero. Note that there is no solution to this set of equations. As a second case, suppose \mathbf{b} were changed to $\begin{bmatrix} 6 \\ 3 \end{bmatrix}$. Here an infinite number of solutions can be obtained, but no unique solution exists.

Degenerate cases such as those above are not frequently encountered. More often, $|\mathbf{A}| \neq 0$. Let

$$\mathbf{A} = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} \quad \text{and} \quad \mathbf{b} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

or

$$2x_1 + x_2 = 1 \quad (\text{A.24a})$$

$$x_1 + 2x_2 = 0 \quad (\text{A.24b})$$

By algebraic substitution, x_1 and x_2 can be found. Multiply Equation (A.24a) by (-0.5) and add this equation to (A.24b),

$$2x_1 + x_2 = 1 \quad (\text{A.24c})$$

$$0 + 1.5x_2 = -0.5 \quad (\text{A.24d})$$

Solve (A.24d) for $x_2 = -0.333$. This result can be substituted into (A.24c) to obtain $x_1 = 0.667$.

The steps employed in Equations (A.24) are equivalent to row operations. The use of row operations to simplify linear algebraic equations is the basis for Gaussian elimination (Golub and Van Loan, 1996). Gaussian elimination transforms the original matrix into upper triangular form, that is, all components of the matrix below the main diagonal are zero. Let us illustrate the process by solving a set of three equations in three unknowns for \mathbf{x} .

EXAMPLE A.8 SOLUTION OF SIMULTANEOUS LINEAR EQUATIONS

Solve for \mathbf{x} given \mathbf{A} and \mathbf{b} .

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 2 & 2 \\ 2 & 1 & 1 \end{bmatrix} \quad \mathbf{b} = \begin{bmatrix} 1 \\ 0 \\ 2 \end{bmatrix} \quad \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

Solution. First a composite matrix from \mathbf{A} and \mathbf{b} is constructed:

$$\mathbf{C}_0 = [\mathbf{A} \mid \mathbf{b}] = \begin{bmatrix} 1 & 0 & 1 & 1 \\ 1 & 2 & 2 & 0 \\ 2 & 1 & 1 & 2 \end{bmatrix}$$

Carry out row operations, keeping the first row intact; successive matrices are

$$\mathbf{C}_1 = \begin{bmatrix} 1 & 0 & 1 & 1 \\ 0 & 2 & 1 & -1 \\ 2 & 1 & 1 & 2 \end{bmatrix} \quad \mathbf{C}_2 = \begin{bmatrix} 1 & 0 & 1 & 1 \\ 0 & 2 & 1 & -1 \\ 0 & 1 & -1 & 0 \end{bmatrix}$$

Next, with the second row in \mathbf{C}_2 kept intact, the upper triangular form is achieved by operating on the third row:

$$\mathbf{C}_3 = \begin{bmatrix} 1 & 0 & 1 & 1 \\ 0 & 2 & 1 & -1 \\ 0 & 0 & -1.5 & 0.5 \end{bmatrix}$$

\mathbf{C}_3 can now be converted to the form of algebraic equations:

$$x_1 + x_3 = 1$$

$$2x_2 + x_3 = -1$$

$$-1.5x_3 = 0.5$$

which can be solved stage by stage starting with the last row to get $x_3 = -0.333$, $x_2 = -0.333$, $x_1 = 1.333$.

Gaussian elimination is a very efficient method for solving n equations in n unknowns, and this algorithm is readily available in many software packages. For solution of linear equations, this method is preferred computationally over the use of the matrix inverse. For hand calculations, Cramer's rule is also popular.

The determinant of \mathbf{A} is unchanged by the row operations used in Gaussian elimination. Take the first three columns of \mathbf{C}_3 above. The determinant is simply the product of the diagonal terms. If none of the diagonal terms are zero when the matrix is reformulated as upper triangular, then $|\mathbf{A}| \neq 0$ and a solution exists. If $|\mathbf{A}| = 0$, there is no solution to the original set of equations.

A set of nonlinear equations can be solved by combining a Taylor series linearization with the linear equation-solving approach discussed above. For solving a single nonlinear equation, $h(x) = 0$, Newton's method applied to a function of a single variable is the well-known iterative procedure

$$x^{k+1} - x^k \equiv \Delta x^k = -\frac{h(x^k)}{dh(x^k)/dx} \quad (\text{A.25})$$

or

$$\left[\frac{dh}{dx} \right]_{x^k} (\Delta x^k) = -h(x^k)$$

where k is the iteration number and Δx^k is the correction to the previous value, x^k . Similarly, a set of nonlinear equations, $\mathbf{h}(\mathbf{x}) = \mathbf{0}$, can be solved iteratively using Newton's method, by solving a set of linearized equations of the form $\mathbf{A}\mathbf{x} = \mathbf{b}$:

$$\left[\frac{\partial \mathbf{h}}{\partial \mathbf{x}} \right]_{\mathbf{x}^k} \cdot \Delta \mathbf{x}^k = -\mathbf{h}(\mathbf{x}^k) \quad (\text{A.26})$$

Note that the Jacobian matrix $\partial \mathbf{h} / \partial \mathbf{x}$ on the left-hand side of Equation (A.26) is analogous to \mathbf{A} in Equation (A.20), and $\Delta \mathbf{x}^k$ is analogous to \mathbf{x} . To compute the correction vector $\Delta \mathbf{x}$, $\partial \mathbf{h} / \partial \mathbf{x}$ must be nonsingular. However, there is no guarantee even then that Newton's method will converge to an \mathbf{x} that satisfies $\mathbf{h}(\mathbf{x}) = 0$.

In solving sets of simultaneous linear equations, the "condition" of the matrix is quite important. If some elements are quite large and some are quite small (but nonzero), numerical roundoff or truncation in a computer can have a significant effect on accuracy of the solution. A type of matrix is referred to as "ill conditioned" if it is nearly singular (equivalent to the scalar division by 0). A common measure of the degree of ill conditioning is the condition number, namely the ratio of the eigenvalues with largest (α_h) and smallest (α_l) modulus:

$$\text{Condition number} = \frac{|\alpha_h|}{|\alpha_l|} \quad (\text{A.27})$$

The bigger the ratio, the worse the conditioning; a value of 1.0 is best. The calculation of eigenvalues are discussed in the next section. In general, as the dimension of the matrix increases, numerical accuracy of the elements is diminished. One technique to solve ill-conditioned sets of equations that has some advantages in speed and accuracy over Gaussian elimination is called "L-U decomposition" (Dongarra et al., 1979; Stewart, 1998), in which the original matrix is decomposed into upper and lower triangular forms.

A.5 EIGENVALUES, EIGENVECTORS

An $n \times n$ matrix has n eigenvalues. We define an n -vector \mathbf{v} , the eigenvector, which is associated with an eigenvalue e such that

$$\mathbf{A}\mathbf{v} = e\mathbf{v} \quad (\text{A.28})$$

Hence the product of the matrix \mathbf{A} multiplying the eigenvector \mathbf{v} is the same as the product obtained by multiplying the vector \mathbf{v} by the scalar eigenvalue e . One eigenvector exists for each of the n eigenvalues. Eigenvalues and eigenvectors provide unambiguous information about the nature of functions used in optimization. If all eigenvalues of \mathbf{A} are positive, then \mathbf{A} is positive-definite. If all $e_i < 0$, then \mathbf{A} is negative-definite. See Chapter 4 for a more complete discussion of definiteness and how it relates to convexity and concavity.

If we rearrange Equation (A.28) (note that the identity matrix must be introduced to maintain conformable matrices),

$$(\mathbf{A} - e\mathbf{I})\mathbf{v} = \mathbf{0} \quad (\text{A.29})$$

$(\mathbf{A} - e\mathbf{I})$ in Equation (A.29) has the unknown variable e subtracted from each diagonal element of \mathbf{A} . Equation (A.29) is a set of linear algebraic equations where \mathbf{v} is

the unknown vector. However, because the right-hand side of (A.29) is zero, either $\mathbf{v} = \mathbf{0}$ (the trivial solution), or a nonunique solution exists. For example in

$$2v_1 + 2v_2 = 0$$

$$v_1 + v_2 = 0$$

then $\det[\mathbf{A}] = 0$, and the solution is nonunique, that is, $v_1 = -v_2$. The equations are redundant. However, if one of the coefficients of v_1 or v_2 in Equation (A.29) changes, then the only solution is $v_1 = v_2 = 0$ (the trivial solution).

The determinant of $(\mathbf{A} - e\mathbf{I})$ must be zero for a nontrivial solution ($\mathbf{v} \neq \mathbf{0}$) to exist. Let us illustrate this idea with a (2×2) matrix:

$$\mathbf{A} = \begin{bmatrix} 1 & 2 \\ 2 & 1 \end{bmatrix} \quad (\mathbf{A} - e\mathbf{I}) = \begin{bmatrix} 1 - e & 2 \\ 2 & 1 - e \end{bmatrix}$$

$$\det \begin{bmatrix} 1 - e & 2 \\ 2 & 1 - e \end{bmatrix} = (1 - e)^2 - 4 = e^2 - 2e - 3 = 0 \quad (\text{A.30})$$

Equation (A.30) determines values of e which yield a nontrivial solution. Factoring (A.30)

$$(e - 3)(e + 1) = 0 \quad e = 3, -1$$

Therefore, the eigenvalues are 3 and -1 . Note that for $e = 3$,

$$\mathbf{A} - e\mathbf{I} = \begin{bmatrix} -2 & 2 \\ 2 & -2 \end{bmatrix}$$

and for $e = -1$,

$$\mathbf{A} - e\mathbf{I} = \begin{bmatrix} 2 & 2 \\ 2 & 2 \end{bmatrix}$$

both of which are singular matrices.

For each eigenvalue there exists a corresponding eigenvector. For $e_1 = 3$, Equation (A.29) becomes

$$\begin{bmatrix} (1 - 3) & 2 \\ 2 & (1 - 3) \end{bmatrix} \begin{bmatrix} v_{11} \\ v_{12} \end{bmatrix} = \mathbf{0}$$

$$-2v_{11} + 2v_{12} = 0$$

$$2v_{11} - 2v_{12} = 0$$

Note that these equations are equivalent and cannot be solved uniquely; the solution to both equations is $v_{11} = v_{12}$. Thus, the eigenvector has direction but not

length. The direction of the eigenvector can be specified by choosing v_{11} and calculating v_{12} . For example, let $v_{11} = 1$. Then

$$\mathbf{v}_1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

The magnitude of \mathbf{v}_1 cannot be determined uniquely. Similarly, for $e_2 = -1$,

$$\mathbf{v}_2 = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

is a solution of (A.29).

For a general $n \times n$ matrix, an n th-order polynomial results from solving $\det(\mathbf{A} - e\mathbf{I}) = 0$. This polynomial will have n roots, and some of the roots may be imaginary numbers. A computer program can be used to generate the polynomial and factor it using a root-finding technique, such as Newton's method. However, more efficient iterative techniques can be found in computer software to calculate both e_i and \mathbf{v}_i (Dongarra et al., 1979).

Principal minors

In Chapter 4 we discuss the definitions of convexity and concavity in terms of eigenvalues; an equivalent definition using determinants of principal minors is also provided. A principal minor of \mathbf{A} of order k is a submatrix found by deleting any $n - k$ columns (and their corresponding rows) from the matrix. The leading principal minor of order k is found by deleting the last $n - k$ columns and rows. In Example A.2, the leading principal minor (order 1) is 1; the leading principal minor (order 2) is $\begin{bmatrix} 1 & 2 \\ 2 & 1 \end{bmatrix}$, and for order 3 the minor is the 3×3 matrix itself.

REFERENCES

- Dongarra, J. J.; J. R. Bunch; C. B. Moler; and G. W. Stewart. *LINPACK User's Guide*, SIAM, Philadelphia, PA (1979).
- Golub, G. H.; and C. F. Van Loan. *Matrix Computations*. Johns Hopkins, Baltimore, MD (1996).
- Meyer, D. C. *Matrix Algebra and Applied Linear Algebra*. SIAM, Philadelphia, PA (2000).
- Stewart, G. W. *Matrix Algorithms: Basic Decomposition*. SIAM, Philadelphia, PA (1998).

SUPPLEMENTARY REFERENCES

- Amundson, N. R., *Mathematic Methods in Chemical Engineering: Matrices and Their Application*. Prentice-Hall, Englewood Cliffs, NJ (1966).
- Campbell, H. G. *An Introduction to Matrices, Vectors, and Linear Programming*. Prentice-Hall, Englewood Cliffs, NJ (1977).

Daniel, J. W. *Applied Linear Algebra*. Prentice-Hall, Englewood Cliffs, NJ (1988).
 Demmel, J. W. *Applied Numerical Linear Algebra*. SIAM, Philadelphia, PA (1997).

PROBLEMS

A.1 For

$$\mathbf{A} = \begin{bmatrix} 1 & 1 \\ 2 & 1 \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} 0 & 4 \\ 1 & 3 \end{bmatrix}$$

Find

- \mathbf{AB} and \mathbf{BA} (compare)
- $\mathbf{A}^T\mathbf{B}$
- $\mathbf{A} + \mathbf{B}$
- $\mathbf{A} - \mathbf{B}$
- $\det \mathbf{A}$, $\det \mathbf{B}$
- $\text{Adj } \mathbf{A}$, $\text{Adj } \mathbf{B}$
- \mathbf{A}^{-1} , \mathbf{B}^{-1} (verify the answer)

A.2 Solve $\mathbf{Ax} = \mathbf{b}$ for \mathbf{x} , where

$$\mathbf{A} = \begin{bmatrix} 1 & 2 & 3 \\ 3 & 2 & 1 \\ 1 & 0 & 1 \end{bmatrix} \quad \mathbf{b} = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix}$$

Use

- Gaussian elimination and demonstrate that \mathbf{A} is nonsingular. Check to see that the determinant does not change after each row operation.
- Use $\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}$.
- Use Cramer's rule.

A.3 Suppose

$$\begin{aligned} z_1 &= 3x_1 + x_3 \\ z_2 &= x_1 + x_2 + x_3 \\ z_3 &= 2x_2 + x_3 \end{aligned} \quad \mathbf{z} = \mathbf{Ax}$$

Find equations for x_1 , x_2 , and x_3 in terms of z_1 , z_2 , z_3 . Use an algebraic method first; check the result using \mathbf{A}^{-1} .

A.4 For

$$\mathbf{x}_1 = \begin{bmatrix} 1 \\ 1 \\ 2 \end{bmatrix} \quad \mathbf{x}_2 = \begin{bmatrix} 1 \\ 0 \\ 2 \end{bmatrix} \quad \mathbf{A} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 4 \end{bmatrix}$$

find the magnitude (norm) of each vector.

What is $\mathbf{x}_1^T\mathbf{x}_2$? $\mathbf{x}_1\mathbf{x}_2^T$? $\mathbf{x}_1^T\mathbf{Ax}_1$?

Find a vector \mathbf{x}_3 that is orthogonal to \mathbf{x}_1 ($\mathbf{x}_1^T\mathbf{x}_3 = 0$). Are \mathbf{x}_1 , \mathbf{x}_2 , and \mathbf{x}_3 linearly independent?

A.5 For

$$\mathbf{A} = \begin{bmatrix} 1 & 1 & 2 \\ -1 & 0 & 1 \\ -2 & -3 & 1 \end{bmatrix}$$

calculate $\det \mathbf{A}$ using expansion by minors of the second row. Repeat with the third column.

A.6 Calculate the eigenvalues and eigenvectors of $\begin{bmatrix} 0 & 1 \\ 1 & 4 \end{bmatrix}$. Repeat for

$$\begin{bmatrix} 2 & 0 & 0 \\ 0 & 3 & 0 \\ 0 & 0 & 2 \end{bmatrix}$$

A.7 Show that for a 2×2 symmetrical matrix, the eigenvalues must be real (do not contain imaginary components). Develop a 2×2 nonsymmetrical matrix which has complex eigenvalues.

A.8 A technique called LU decomposition can be used to solve sets of linear algebraic equations. \mathbf{L} and \mathbf{U} are lower and upper triangular matrices, respectively. A lower triangular matrix has zeros above the main diagonal; an upper triangular matrix has zeros below the main diagonal. Any matrix \mathbf{A} can be formed by the product of \mathbf{LU} .

(a) For

$$\mathbf{A} = \begin{bmatrix} 1 & 1 & 0 \\ 2 & 3 & 1 \\ 1 & 0 & 1 \end{bmatrix}$$

find some \mathbf{L} and \mathbf{U} that satisfy $\mathbf{LU} = \mathbf{A}$.

(b) If $\mathbf{Ax} = \mathbf{b}$, $\mathbf{LUx} = \mathbf{b}$ or $\mathbf{Ux} = \mathbf{L}^{-1}\mathbf{b} = \hat{\mathbf{b}}$.

Let

$$\mathbf{b} = \begin{bmatrix} 1 \\ 1 \\ 2 \end{bmatrix}$$

Calculate \mathbf{L}^{-1} and $\hat{\mathbf{b}}$. Then solve for \mathbf{x} using substitution from the upper triangular matrix \mathbf{U} .

A.9 You are to solve the two nonlinear equations,

$$x_1^2 + x_2^2 = 8$$

$$x_1x_2 = 4$$

using the Newton–Raphson method. Suggested starting points are (0, 1) and (4, 4).

APPENDIX B

COST ESTIMATION

B.1	Capital Costs	604
B.2	Operating Costs	610
B.3	Taking Account of Inflation	611
B.4	Predicting Revenues in an Economic-Based Objective Function	614
B.5	Project Evaluation	615
	References	628

IN CHAPTER 3 we discussed the formulation of objective functions without going into much detail about how the terms in an objective function are obtained in practice. The purpose of this appendix is to provide some brief information that can be used to obtain the coefficients in objective functions in economic optimization problems. Various methods and sources of information are outlined that help establish values for the revenues and costs involved in practical problems in design and operations. After we describe ways of estimating capital costs, operating costs, and revenues, we look at the matter of project evaluation and discuss the many contributions that make up the net income from a project, including interest, depreciation, and taxes. Cash flow is distinguished from income. Finally, some examples illustrate the application of the basic principles.

The estimation of operating and capital costs is an important facet of process design and optimization. In the absence of firm bids or valid historical records, you can locate charts, tables, and equations that provide cost estimates from a wide variety of sources based on given values of the design variables.

1. Specialized books on cost estimation such as Garrett (1989) or Ostwald (1992).
2. Textbooks on plant design such as Turton et al. (1998) and Seider et al. (1999).
3. Handbooks such as *Perry's Chemical Engineering Handbook* (Green and Maloney, 1997).
4. Trade magazines such as *Chemical Engineering* or the *Oil and Gas Journal*.
5. Literature provided by equipment vendors.
6. Reports and books published by professional societies.
7. Local, state, and federal government publications.
8. Databases in process simulators such as Aspen (1998), HYSYS (1998), and ProII (1998).
9. The Internet (<http://www.chempute.com> or <http://www.chemengineer.miningco.com>).
10. Commercial software for process equipment cost estimation such as CHEM-COST (Icarus Corp., 1999).

The preceding listed sources provide information on current and often historical capital and operating costs that can be used in your current and projected economic evaluation.

B.1 CAPITAL COSTS

In carrying out an economic analysis, recognize that various levels of detail in the design of a process exist.

1. Rough feasibility estimate based on a general flowsheet using historical costs, charts, or the literature and using multiplying factors based on experience to scale for inflation, size differences, and tax rates. Examine Figure B.1 for cost estimates based on entire plants as a function of capacity.

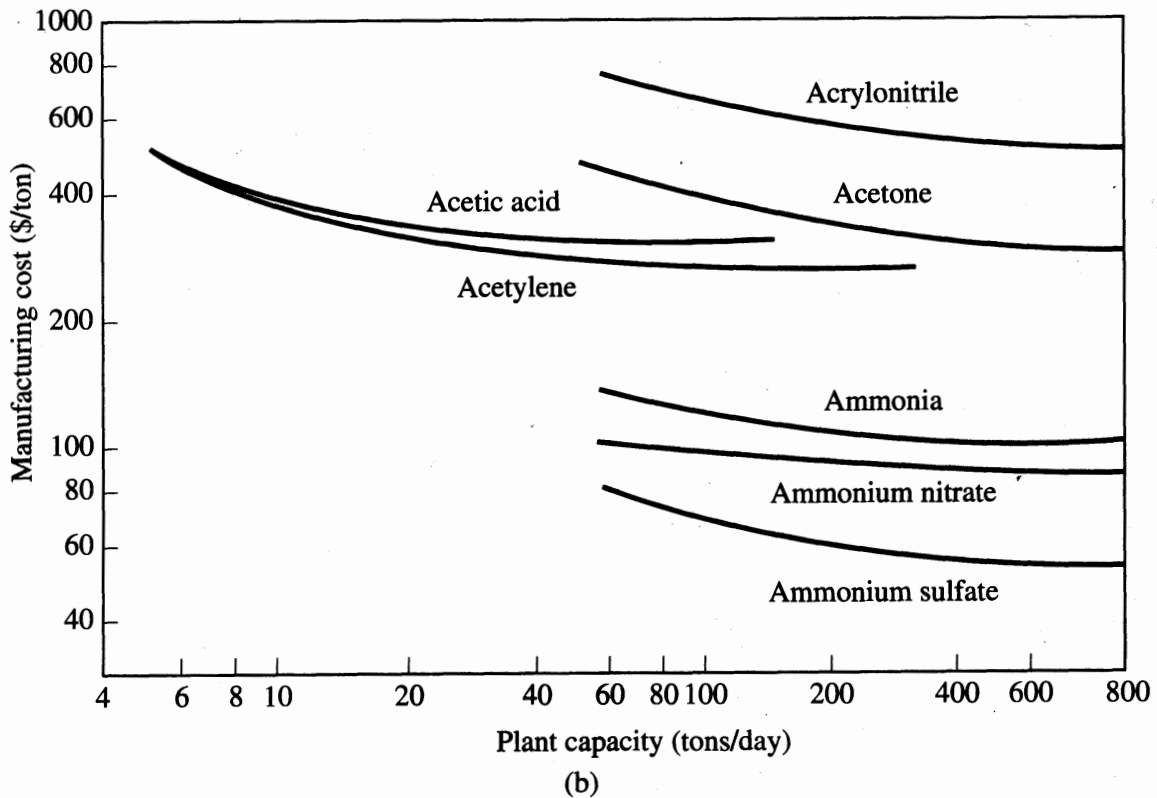
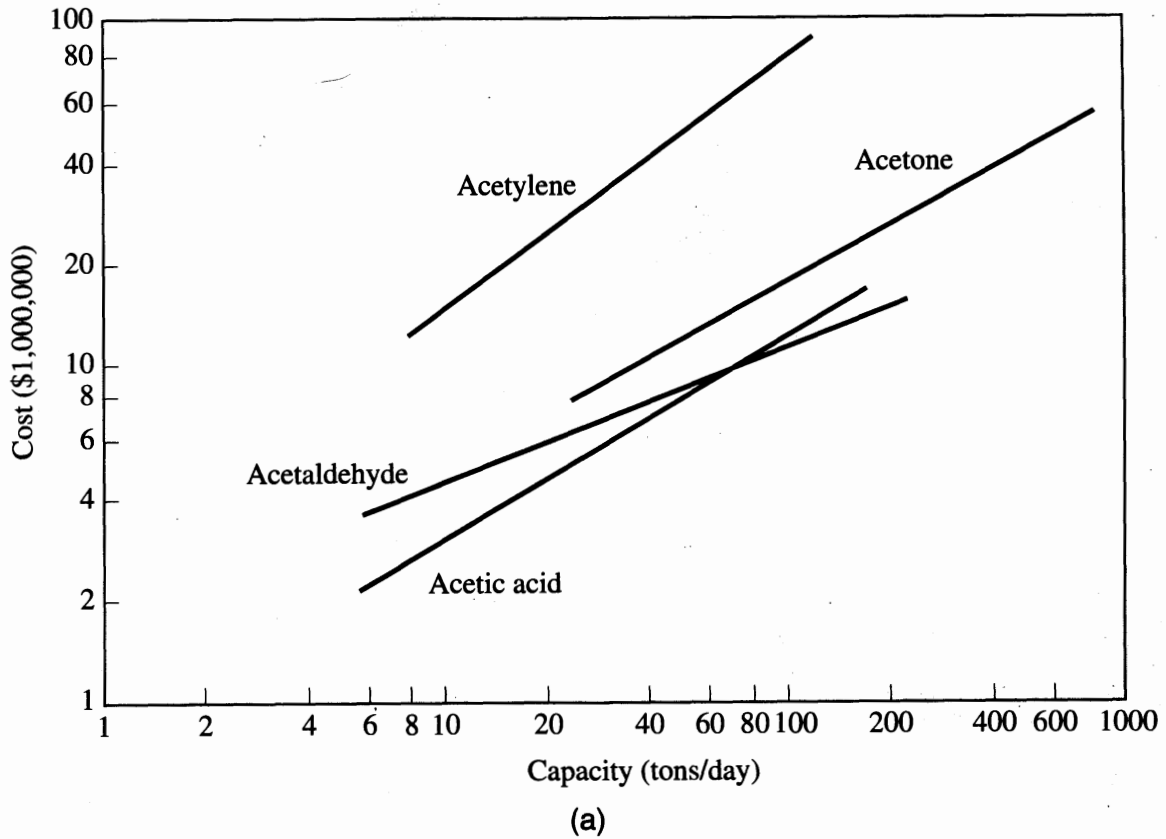


FIGURE B.1
 Rough estimates of (a) complete plant costs and (b) manufacturing costs (in tons/day) based on historical data (from Garrett, 1989 with permission from Kluwer Academic Publishers).

Purchased cost of tank, delivered basis: \$100.00

Installation costs:

Piping	12.8	
Concrete	8.6	
Instruments	3.8	
Electrical	0.6	
Paint	1.2	<u>27.0</u>
	Total materials	<u>127.0</u>

Labor costs:

Man hours/\$ materials: 0.044

Average hourly labor costs
including fringe: \$28.50

(0.044) (127.0) (\$28.50) =	<u>159.3</u>
Total	<u>286.3</u>

Indirect (overhead) cost factor: 1.26

1.26 (286.3) = 360.7

Total cost	<u>360.7</u>
------------	--------------

FIGURE B.2

Approximate relative proportions of the cost of a 30,000-gallon tank erected in the field (1 unit of 12 in the flowsheet).

2. Major equipment estimates based on a more detailed given flowsheet that includes all of the equipment of significance roughly sized with approximate costs. Optimization using process flow simulators (refer to Chapter 15) can be employed. Figure B.2 illustrates a typical analysis for a tank. Refer to Brown (2000) for additional details.
3. Confirmed design in which additional detail and costs are developed for the arrangement of equipment, piping, utilities (water, steam, electrical, air), instrumentation, and control systems.
4. Final design that provides the plans, specifications for all equipment, detailed sections for the flowsheets, quotes from vendors, inhouse budgets, and a schedule for implementation.

As you may well surmise, more approximate designs lead to larger error bounds, running from perhaps $\pm 50\%$ of the total cost for category 1 to $\pm 5\%$ for category 4. The cost of making the estimates, of course, increases as the extent of the information about the design increases. A very preliminary design might cost from \$5000 to \$10,000, whereas the final design runs from 1% to as much as 5% of the total plant cost. Process simulators (refer to Chapter 15) make the preliminary stages of a design fairly easy to implement.

TABLE B.1
Components to include in capital
cost estimation

Purchased equipment such as	
Towers, columns	Boilers
Reactors	Generators
Heat exchanger	Air conditioning
Cooling towers	Refrigeration
Tanks	
Piping	
Electrical	
Instrumentation	
Utilities such as	
Power	
Water	
Sewage, waste handling	
Electricity	
Insulation	
Buildings (and possibly land)	
Installation costs such as	
Labor	
Painting	
Fireproofing	
Supervision	
Inspection	
Safety, fire fighting	
Engineering, design, licensing	
Laboratory	
Shipping (working capital start-up expenses)	

What components must be included in estimating plant capital costs? Table B.1 is a partial list with some specific details.

Charts, correlations, and tables in the sources cited earlier relate capital costs to various parameters characteristic of the equipment to be evaluated. Table B.2 lists typical parameters used to correlate equipment costs for common types of process equipment. Figure B.3 is an example of such correlations for the cost of heat exchangers as a function of exchanger area. These forms of cost curves generally appear as nearly straight lines on log-log plots, indicating a power-law relationship between capital cost and capacity, with exponents typically ranging from 0.5 to 0.8.

If you want to scale up or scale down process equipment using one of the parameters in Table B.2, a typical rule of thumb is the following relation

$$\log C_B = a_1 + a_2 \log S \quad (\text{B.1})$$

where C_B = base cost

S = size parameter

a_1, a_2 = coefficients to be estimated from valid data

TABLE B.2
Process parameters used in cost estimation for typical process equipment

Equipment type	Economic variables
1. Flashdrum	Diameter, height, material of construction, internal pressure
2. Distillation column, tray absorber	Diameter, height, internal pressure, material of construction, tray type, number of trays, condenser, reboiler (see item 3)
3. Condenser, reboiler, heat exchanger (shell and tube)	Heat transfer surface area, type, shell design pressure, materials for shell and tube
4. Absorber (packed)	Diameter, height, internal pressure, material of construction, packing type, packing volume
5. Process furnace or direct-fired heater	Design type, absorbed heat duty, pressure, tube material, capacity
6. Pumps (centrifugal, reciprocating)	Fluid density, capacity, dynamic head, type, driver, operating condition limits, material of construction
7. Gas compressor	Brake horsepower, driver type
8. Storage tank	Tank capacity, type, and storage pressure
9. Boiler	Steam flow rate, design pressure, steam superheat
10. Reactor	Type, diameter, height, design pressure, material of construction, capacity

A base cost typically corresponds to carbon steel construction and pressure below 100 psi. Note that Equation (B.1) is equivalent to

$$C_B = a' x_1 S^{a_2} \quad (\text{B.2})$$

the familiar formula for scale-up, where a_2 is typically about 0.6. A slightly different correlation provides a more accurate fit of cost data by using three coefficients.

$$\log C_B = a_1 + a_2 \log S + a_3 (\log S)^2 \quad (\text{B.3})$$

The estimated capital cost C_E for equipment can be found from base cost C_B from

$$C_E = C_B f_D f_M f_p \quad (\text{B.4})$$

where f_D = design type cost factor

f_M = material of construction cost factor

f_p = pressure rating factor

The design type refers to variations in equipment configuration (e.g., fixed head versus floating head in a heat exchanger). The adjustment for material of construction is used principally to account for the use of alloy steel instead of carbon steel. The pressure rating factor allows adjusting costs for pressures other than the refer-

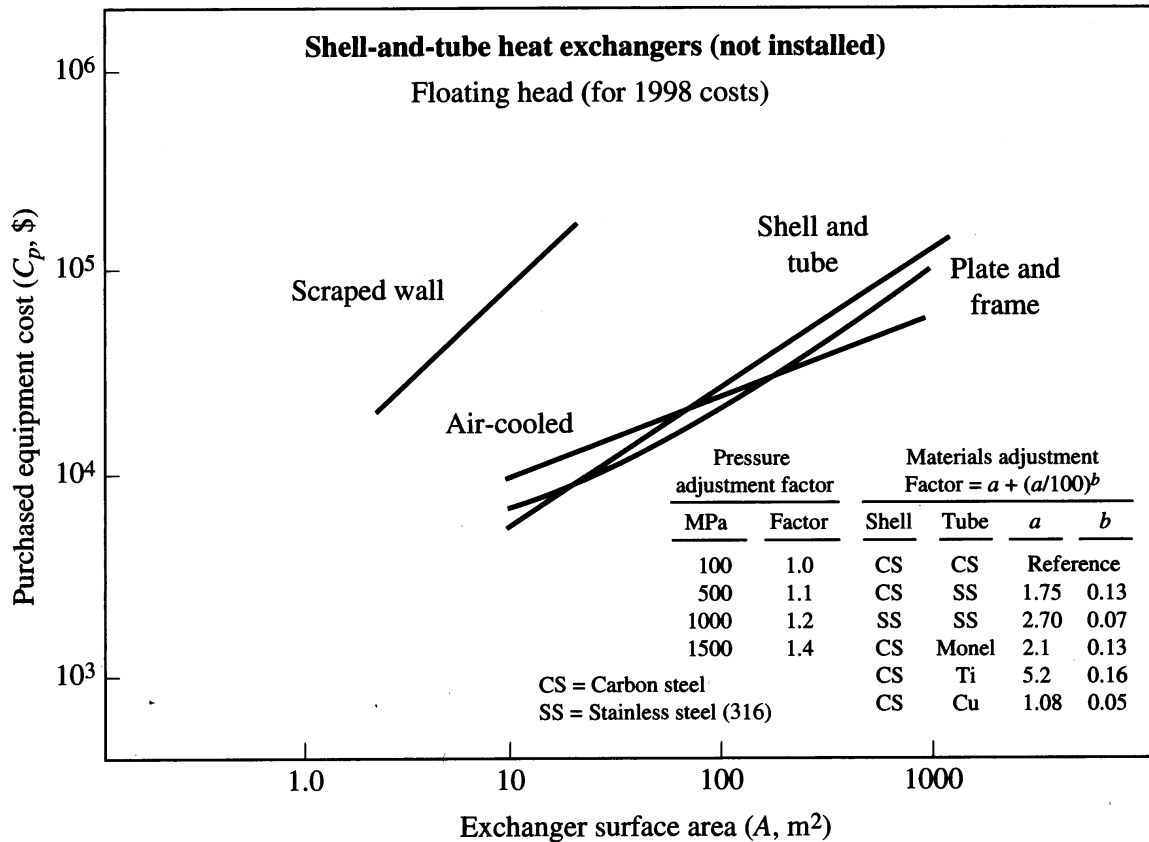


FIGURE B.3
Purchased equipment costs for various types of heat exchangers.

ence pressure. Obviously, higher pressure operation causes additional capital costs because of thicker vessel walls, and so on; f_p may be a discontinuous function.

EXAMPLE B.1 CAPITAL COST ESTIMATION

Suppose the cost for a fixed-head heat exchanger constructed of 316 stainless steel operating at 300–600 psi is to be estimated. The base case is a carbon steel, floating-head exchanger operating at 100 psi of area A . For such operation (Kuri and Corripio 1984), the base cost is

$$C_B = \exp[8.551 - 0.30863(\ln A) + 0.06811(\ln A)^2] \tag{a}$$

where A is the exchanger heat transfer area in square feet ($150 \leq A \leq 12,000 \text{ ft}^2$) and C_B is in dollars. Multiply C_B by factors f_D , f_p , and f_M , calculated as follows:

For a fixed head (versus floating head)

$$f_D = \exp[-1.1156 + 0.0906(\ln A)] \tag{b}$$

For 300 to 600 psi, the correction is

$$f_p = 1.0305 + 0.07140(\ln A) \tag{c}$$

For 316 stainless steel, the correction is

$$f_M = 2.7 \quad (d)$$

Equation (B.4) can then be used to determine the actual capital cost for a specified area A.

For equipment such as distillation columns, the costs of several components (trays, shell) must be calculated.

B.2 OPERATING COSTS

In carrying out an economic evaluation of a proposed process or a modification of an existing one, estimation of future operating costs is just as important as estimating the capital costs involved in the analysis.

Operating costs include the costs of raw materials, direct operating labor, labor supervision, maintenance, plant supplies, utilities (steam, gas, electricity, fuel), property taxes, and insurance. Sometimes certain operating cost components are directly expressed as a fraction of the capital investment cost. Table B.3 is a brief checklist

TABLE B.3
Preliminary operating cost estimates

- A. Direct production cost
 - 1. Materials
 - a. Raw materials: estimate from price lists, government and trade group reports
 - b. Byproduct and scrap credit: estimate from price lists
 - 2. Utilities: from literature or similar operations
 - 3. Labor: from historical data, manning tables, literature, or similar operations
 - 4. Supervision: 10–25% of labor
 - 5. Fringe benefits and FICA: 30–45% of labor plus supervision
 - 6. Maintenance: 2–10% of investment per year
 - 7. Operating supplies: 0.5–1.0% of investment per year, or 6–10% of operating labor
 - 8. Laboratory: 10–20% of labor per year
 - 9. Waste disposal: from literature, similar operations, or separate estimate
 - 10. Royalties: 1–5% of sales
 - 11. Contingencies: 1–5% of sales
- B. Indirect costs
 - 1. Depreciation: 10–20% of investment per year
 - 2. Real estate taxes: 1–2% of investment per year
 - 3. Insurance: 0.5–1.0% of investment per year
 - 4. Interest: 10–12% of investment per year
 - 5. General administrative overhead: 50–70% of labor, supervision, and maintenance, or 6–10% of sales
- C. Distribution costs
 - 1. Packaging: estimate from container costs
 - 2. Shipping: from carriers or 1–3% of sales

TABLE B.4
Rates for industrial utilities, 1998

Utility	Cost (\$)	Unit
Steam		
500 psi (250°C)	8.00–9.00	1000 kg
(200°C)	6.00–8.00	1000 kg
Exhaust (100°C)	5.00–7.00	1000 kg
Electricity		
Purchased	0.03–0.08	kWh
Self-generated	0.02–0.06	kWh
Cooling water (30°C)		
Well	8.6–46	1000 m ³
River or salt	6.0–17	1000 m ³
Tower	6.0–8.0	1000 m ³
Process water		
City	5.00–8.00	1000 m ³
Boiler feed	1.70–2.70	1000 m ³
Compressed air		
Process air	1.60–4.80	1000 m ³
Instrument	3.20–10.00	1000 m ³
Natural gas	2.00–4.00	10 ⁶ Btu
Fuel oil	0.30–0.50	gal
Coal	4.00–5.00	mton
Refrigeration (–30°C)	2.00–3.00	ton/day (288,000 Btu removed)

for estimating operating costs; note that such items as property taxes, insurance, and maintenance are computed as fractions of total fixed capital investment.

You may wonder how you can determine operating costs for a plant or process that is not yet operating. In Table B.4 you will note various rules of thumb that can be used to compile specific categories of approximate operating costs. If more detail is needed and if the appropriate information is not in your existing databases, then you can refer to some of the sources cited at the beginning of this chapter. For example, to collect more detailed information on utility costs you could prepare a table such as Table B.4 from data in financial newspapers and the Internet. As another example, detailed labor costs for operators can be assembled by considering the number of operators per shift for a section of the plant or piece of equipment, the number of days you expect to operate per year, the number of shifts per day, the expected average wage per operator to which have to be added fringe benefits and FICA taxes. Raw materials costs are available from bids, the *Chemical Marketing Reporter*, or the *Chemical Buyer's Guide*. Operating costs can vary from location to location so you should obtain local data whenever possible.

B.3 TAKING ACCOUNT OF INFLATION

Frequently you can find cost data that are appropriate for your economic evaluation, but they may be out of date. By taking account of the inflation in cost you can escalate old costs to current values and project current (or old) costs into the future.

Figure B.4 displays four well-known cost indexes for capital costs from 1950 to 1999:

1. ENR: *Engineering News-Record Construction and Building Indexes*
2. CE: *Chemical Engineering Plant Cost Index*
3. M & S: *Marshall and Swift Equipment Index* (also appears in *Chemical Engineering*)
4. NRC: *Nelson–Farrar Refinery Construction Index* (appears in *Oil and Gas Journal*)

Note that from 1950 to 1965–1970, the slopes (except the CE plant cost) of the indices were similar, that the slopes increased substantially during the inflationary period from 1965–1970 to about 1985; thereafter they returned roughly to their original values of about 6 percent per year.

If you need historical values for the cost of specific types of equipment, materials, fuels, and so on, rather than a general index, consult the references cited at the beginning of the chapter. To determine capital costs (C_x) in the year X in the future, given a known cost C_y in year Y , you simply multiply C_y by the ratio of the index (I_x/I_y):

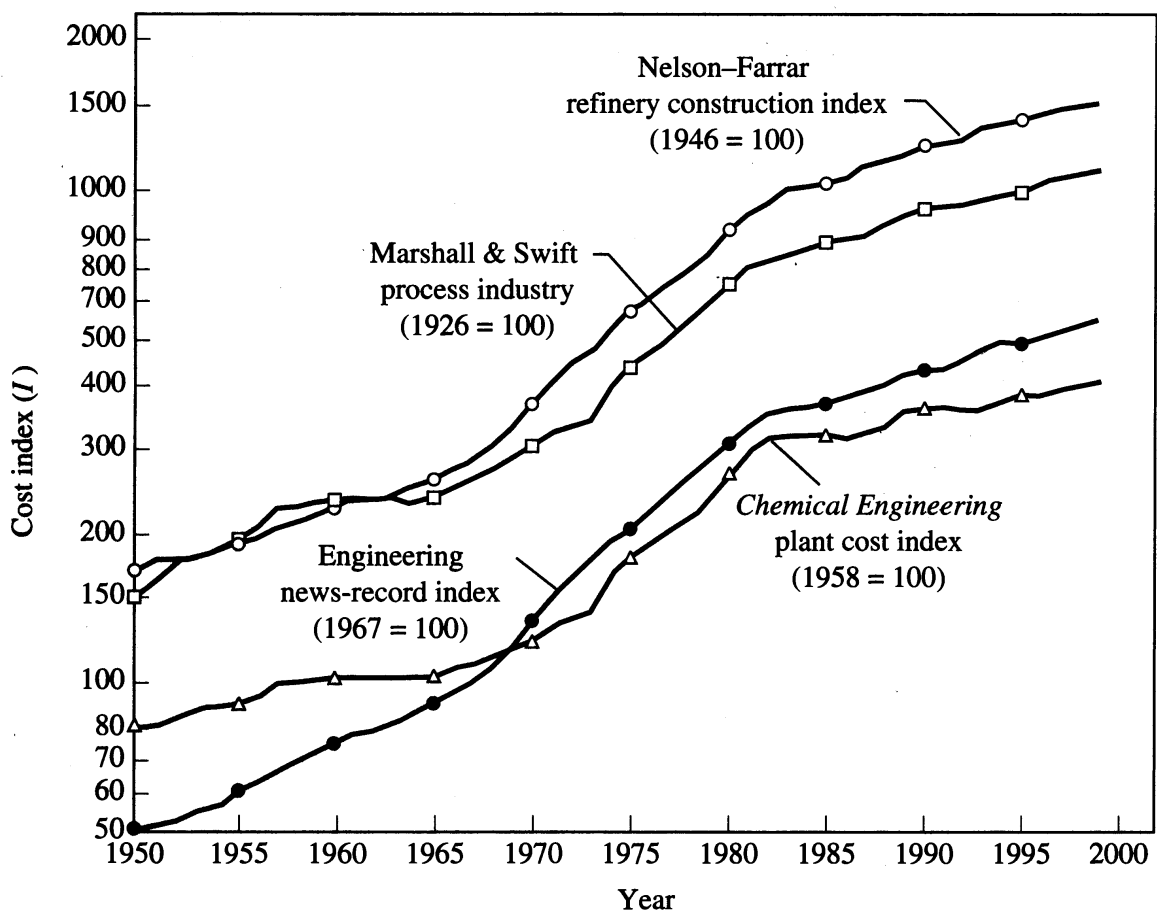


FIGURE B.4

History of selected cost indexes pertinent to chemical process construction (1950–1998)

$$C_x = C_y \cdot \frac{I_x}{I_y} \quad (\text{B.5})$$

The U.S. Bureau of Labor Statistics provides information that permits computation of estimated future labor and material costs. You can project costs to the future by fitting a cost index values for several time periods. If the slope b of the index is constant, then the ratio I_x/I_y versus t is a semilog plot

$$\ln\left(\frac{I_x}{I_y}\right) = bt \quad (\text{B.6})$$

Labor costs experience inflation just as do capital costs as Figure B.5 demonstrates. Raw materials and fuel costs are subject to considerable erratic fluctuations as demonstrated by oil and metals prices, which have rapidly risen and fallen several times over the last five decades. For example, Figure B.6 shows the changes in refinery fuel price index since 1955. Prediction of refinery fuel prices in the future is clearly much more difficult than predicting capital costs.

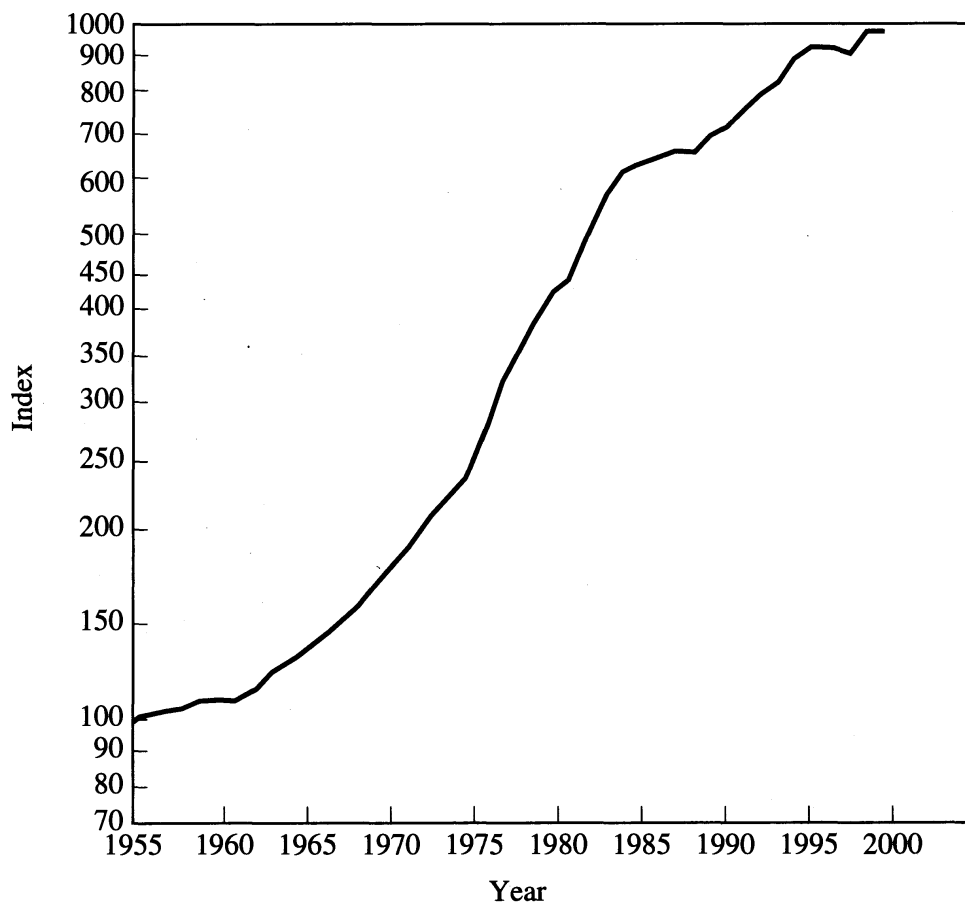


FIGURE B.5

Nelson-Farrar index of operating labor cost (wages plus benefits)
1955-1999 (1956 = 100).

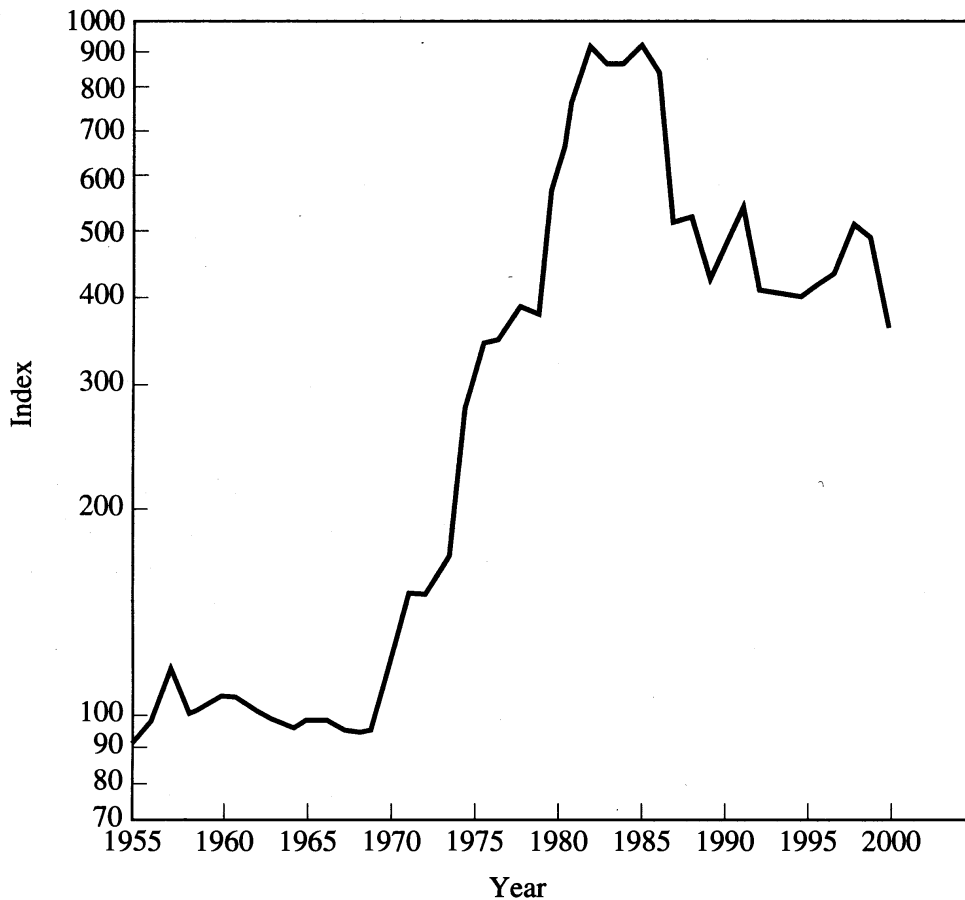


FIGURE B.6

Nelson-Farrar index of refinery fuel: 1955-1999 (1956 = 100).

B.4 PREDICTING REVENUES IN AN ECONOMIC-BASED OBJECTIVE FUNCTION

In maximizing profits over future periods, you have to estimate revenues along with costs. Revenues involve both quantities sold and their prices. The *top-down approach* involves disaggregation, namely starting with estimates of revenues of an entire industry or specialized market that includes the categories of products using company economic models or predictions by industrial trade associations. Then you estimate your company's share of each category. Next, you estimate revenues for a specific product in the category and estimate your company's share for the specific product. The categories can be nested within each other by sales territory, distributor, salesperson, and so on.

The other approach is the *bottom-up procedure*, which proceeds to aggregate projected sales data. You start with the projected sales data in each territory for each product and sum up the forecasts into successively larger amalgamations.

Forecasting revenues fundamentally rests on models plus judgment. More formal methods project the trends of past revenues into the future adjusting for known or expected fluctuations. Typical models employed are

1. Time series
2. Moving averages and smoothing
3. Regression
4. Kalman filters
5. Stochastic models
6. Error models
7. Neural nets

and are adjusted periodically based on the available data. Data can be historical in your database or taken from the reference cited at the start of this chapter. Keep in mind that estimates of future revenues have greater uncertainty than estimates of capital and operating costs. Look at Figure B.6 and imagine you were selling refinery fuel rather than buying it. How much error would occur in predictions of price made in 1969? 1980? 1988? Although sales volume changes with price to some extent, severe price fluctuations are more likely to occur than severe quantity fluctuations. In forecasting, expect unexpected disturbances and allow a margin for error in terms of probability distributions or “worst case” scenarios.

B.5 PROJECT EVALUATION

In Chapter 3, we discussed several criteria involving profitability including:

Payback period (PBP): the cost of an investment divided by the cash flow per period.

Net present value (NPV): the present value (including the time value of money) of initial and future cash flows given by Equation (13.4).

Internal rate of return (IRR): the interest or discount rate for which the future net cash flows equal the initial cash outlay.

Table 3.2 compared the respective features of these three criteria, and in the next two examples we illustrate the specific calculations involved in evaluating projects.

EXAMPLE B.2 USE OF PBP, NPV, AND IRR TO EVALUATE TWO POTENTIAL PROJECTS

Two alternative projects are under consideration. Project A has a project life of 10 years and requires an initial investment of \$100,000 with an annual cash flow after taxes of \$20,000/year for each of 4 years followed by \$10,000/year for years five

through ten. Project *B* has a life of 10 years and requires the same investment but has cash flows of \$15,000/year for each year. Based on the information presented in Chapter 3, evaluate projects *A* and *B* using (a) payback period, (b) internal rate of return, and (c) net present value, assuming an interest rate of 10 percent ($i = 0.10$).

Solution

(a) The respective payback periods are

Project A. It requires 4 years @ \$20,000 plus 2 years @ \$10,000, or a total of 6 years to recover the investment.

Project B.

$$\frac{\$100,000}{\$15,000} = 6.67 \text{ years}$$

These payback periods are quite close.

(b) To find the NPV of the two projects we calculate using Equation (3.4).

Project A.

$$\begin{aligned} \text{NPV} = & -\frac{100,000}{(1 + 0.10)^0} + \frac{20,000}{(1 + 0.10)^1} + \frac{20,000}{(1 + 0.10)^2} \\ & + \frac{20,000}{(1 + 0.10)^3} + \frac{20,000}{(1 + 0.10)^4} + \sum_{k=5}^{10} \frac{10,000}{(1 + 0.10)^k} = -\$7,128.67 \end{aligned}$$

Project B.

$$\text{NPV} = -\frac{100,000}{(1 + 0.10)^0} + \sum_{k=1}^{10} \frac{15,000}{(1 + 0.10)^k} = -\$7,831.49$$

Again the values are quite close.

(c) To find the IRR of the two projects we calculate i with NPV = 0 using Equation (3.4).

Project A.

$$\begin{aligned} 0 = & -\frac{100,000}{(1 + i)^0} + 20,000 \sum_{k=1}^4 \frac{1}{(1 + i)^k} \\ & + 10,000 \sum_{k=5}^{10} \frac{1}{(1 + i)^k}, \quad \text{the solution is } i = 8.06\% \text{ annually} \end{aligned}$$

Project B.

$$0 = -\frac{100,000}{(1 + i)^0} + 15,000 \sum_{k=1}^{10} \frac{1}{(1 + i)^k}$$

The solution is $i = 8.14\%$ annually.

Presumably, neither of the projects would be favorable. Calculations such as made in this example engender a high degree of uncertainty because of the long periods involved, so that a decision between projects, if implemented, is a toss-up.

NPV does not require that the total lives (or multiples thereof) of projects be equal for a comparison to be made. Thus, ambiguous and sometimes contradictory results can arise in using IRR versus NPV [Brigham (1982), Woinsky (1996)]. Jelen and Black (1983) have suggested a comparison based on uniform annual cost, called *unacost*.

EXAMPLE B.3 CALCULATION OF IRR AND NPV FOR PROJECTS WITH DIFFERENT LIFETIMES

Suppose project *C* has a 20-year life and a yearly after-tax cash flow of \$48,000 for an initial investment of \$300,000. Project *D* has a 5-year life, with a yearly cash flow of \$110,000 for an initial investment of \$300,000. Compare the internal rate of return and net present value (for $i = 0.08$) for each option.

Solution. Because the annual cash flows are uniform for projects *C* and *D*, we can apply Equation (3.4a). The internal rates of return are $i_C = 0.15$ for project *C* and $i_D = 0.25$ for project *D*. The advantage of project *D* is a more concentrated period of early cash generation at a high level. For a value of $i = 0.08$, the NPV of each project is as follows:

Project C:

$$\begin{aligned} \text{NPV} &= \left(\sum_{j=1}^{20} \frac{48,000}{(1+i)^j} \right) - 300,000 \\ &= 470,600 - 300,000 = \$170,600 \end{aligned}$$

Project D:

$$\begin{aligned} \text{NPV} &= \left(\sum_{j=1}^5 \frac{110,000}{(1+i)^j} \right) - 300,000 \\ &= 438,200 - 300,000 = \$138,200 \end{aligned}$$

Therefore, based strictly on this calculation, project *C* would be favored over *D* because over its lifetime (20 years versus 5 years), it would generate more (discounted) cash flow. This conclusion is in conflict, however, with that obtained by comparing the IRRs of the two projects. The ranking based on NPV may change if a

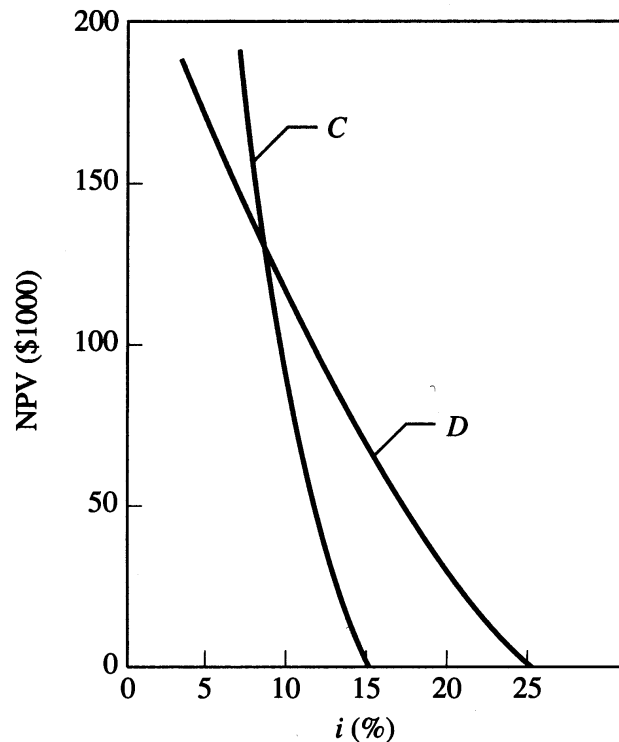


FIGURE EB.3

Comparison of the net present value (NPV) for two projects as a function of i .

different interest rate is assumed. Figure EB.3 shows how NPV varies for each project as a function of i (note the crossover point). Brigham (1982) has concluded that the use of NPV is preferable to IRR, because NPV gives more realistic results for a wide variety of cases, especially when cash flows vary greatly from year to year.

One important assumption to keep in mind in the calculations outlined earlier is that the interest rate (discount rate) has been assumed to be constant over time even though it is not in practice. Examine Figure B.7, which shows how the interest rate for U.S. Treasury securities has changed over time for various durations of investment ranging from 3 months to 30 years (called the yield curve).

To make a decision about investing in a project, more than just cash flows need to be taken into account. Cash flows are reasonably clear-cut, whereas using earnings as a criterion in a multiyear project involves a number of accounting and legal decisions that affect the profitability.

To distinguish between cash flows and earnings, let us look at a grossly simplified set of financial statements for a company. The three statements are a

1. Balance sheet
2. Cash flow statement
3. Income and expense statement

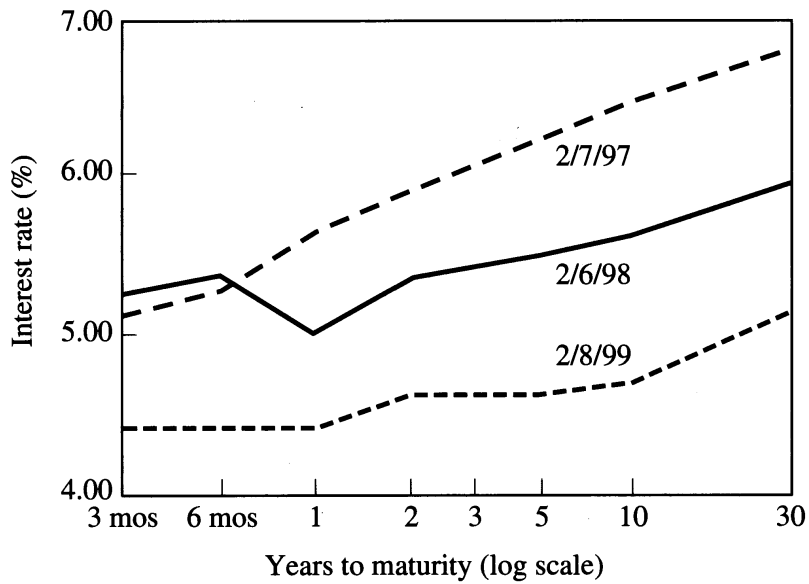


FIGURE B.7
Interest rate provided by U.S. Treasury bills, notes, and bonds at different dates.

Balance Sheet	
Assets:	
Cash	\$100,000
Building	900,000
Total	\$1,000,000
Liabilities and equities:	
Long-term debt	\$600,000
Equity	400,000
Total	\$1,000,000

(Must always be equal)

FIGURE B.8
A simplified balance sheet.

Figure B.8 illustrates the balance sheet. A balance sheet is a snapshot of the assets and liabilities at one point in time. It tells nothing about the transactions and adjustments that led to the numbers presented in the statement. A comparison of balance sheets over time can help indicate earnings.

Next, Figure B.9 represents a simplified cash flow statement for a retail computer store. The bottom number in the statement does not represent profit (income, earnings)—just the net of the cash flows, because the \$30,000 mortgage payment

Cash Flow Statement	
Receipts from sales during the year	\$180,000
Disbursements during the year:	
Maintenance	\$(10,000)
Property taxes	(30,000)
Mortgage payments to principal	(30,000)
Mortgage payments, to interest	(60,000)
	<hr/>
Net before taxes	\$50,000
Less income tax	(10,000)
	<hr/>
Cash left after paying taxes	\$40,000

FIGURE B.9

A simplified cash flow statement.

applied to principal is not deemed to be an item of expense, and the statement does not include a noncash expense incurred for depreciation of \$20,000.

The third statement shown in Figure B.10 is for income and expense that leads to net after-tax profits (earnings), a quantity that transfers to the balance sheet periodically in the category called equity.

Figure B.10 gives you the correct \$50,000 “bottom line.” Note that both depreciation and interest are listed as deductible expenses. Interest is clearly an expense; but why depreciation? Unlike interest and other expense deductions, depreciation does not actually reduce operating cash. Nevertheless, we know that aging and obsolescence over a period of years does decrease the value of most things; depreciation is a loss. So you subtract it from income as you do for a cash expenditure.

The reconciliation between the cash flow statement and the income and expense statement is as follows. Start with the \$40,000 from the last line in the cash flow statement, subtract \$20,000 for the depreciation expense, and add back the \$30,000 mortgage loan principal payment (not an allowed expense). The result is the net after-tax earnings. Figure B.11 is a set of statements from a small oil company. The statement of operations lists revenue and expenses, whereas the balance sheet lists various assets, liabilities, and stockholders’ equity (“net worth”). So-called capital items such as buildings, equipment, oil and gas property, and various intangibles are assets. Operating costs are deductions from revenues for operations not including expenditures for capital items.

Some of the categories and terms on the statement require brief additional explanation.

Income and Expense Statement	
Income from sales	\$180,000
Expenses	
Maintenance, property taxes	\$ 10,000
Property taxes	30,000
Interest	60,000
Depreciation	<u>20,000</u>
Total expenses	<u>120,000</u>
Before tax earnings	<u>\$ 60,000</u>
Income tax on \$60,000 of earnings	<u>10,000</u>
Net after-tax earnings	<u>\$ 50,000</u>

FIGURE B.10

A simplified statement of income and expenses.

Revenues

Revenues include cash received from sales of products and services. Cash received from the sale of equipment, buildings, and equipment is not considered revenue but is instead a decrease in the property and equipment accounts (assets).

Operating expenses

These cash expenses are those necessary to carry on the business, that is, expenses paid to generate revenue. A capital expenditure for plant or equipment generally is not an expense but an addition to the plant or equipment account (an asset). Typical expenses include cost of products sold, repairs, insurance, salaries, property taxes, and so on.

General and administrative expense

These are expenses that are not directly attributed to products, services, or plant, or equipment, such as legal fees, corporate salaries, research expenditures, charitable contributions, and so on.

Interest

Interest paid on loans and mortgages is usually segregated from other expenses.

Statements of Operations (Unaudited)	<i>Three Months Ended September 30,</i>		<i>Nine Months Ended September 30,</i>	
	2000	1999	2000	1999
	Revenues.....	\$ 3,724,004	\$ 2,745,590	\$ 9,927,736
Operating expenses.....	(1,898,765)	(1,163,249)	(4,907,689)	(3,306,535)
Gross margin.....	1,825,239	1,582,341	5,020,047	4,145,451
General and administrative expense.....	(489,843)	(597,905)	(1,405,316)	(1,722,717)
Interest.....	(235,645)	(182,192)	(734,376)	(409,974)
Minority interest.....	2,919	—	17,854	75,086
Income before depletion, depreciation and amortization.....	1,102,670	802,244	2,898,209	2,087,846
Depletion, depreciation and amortization...	(858,534)	(650,492)	(2,275,608)	(1,699,073)
Income before taxes.....	244,136	151,752	622,601	388,773
Income taxes.....	(53,154)	(33,825)	(139,754)	(86,025)
Net income.....	<u>\$ 190,982</u>	<u>\$ 117,927</u>	<u>\$ 482,847</u>	<u>\$ 302,748</u>
Balance Sheets (Unaudited)	<i>September 30, 2000</i>	<i>December 31, 1999</i>		
Assets				
Current assets.....	\$ 5,328,619	\$ 4,753,476		
Notes receivable and investments.....	—	1,403,640		
Oil and gas properties, net.....	17,797,004	16,260,990		
Property and equipment, net.....	3,267,741	1,913,897		
	<u>\$ 26,393,364</u>	<u>\$ 24,332,003</u>		
Liabilities and stockholders' equity				
Current liabilities.....	\$ 5,313,891	\$4,681,323		
Senior debt.....	10,493,784	9,565,428		
Subordinated notes.....	1,753,400	2,123,188		
Minority interest.....	72,888	—		
Stockholders' equity.....	8,759,401	7,962,064		
	<u>\$ 26,393,364</u>	<u>\$ 24,332,003</u>		

FIGURE B.11
Statement of a small oil company.

Depletion

Depletion is noncash allowance deductible from revenue for the recovery of the costs of a natural resource such as oil, gas, coal, or timber. The concept is that as the natural resource is exhausted, the assets of the company are depleted.

Amortization

Amortization is the recovery of certain capital expenditures that can be deducted from revenue in a manner similar to depreciation (discussed in the next section). Typical capital expenditures that can be amortized are pollution control facilities, removal of architectural barriers for the handicapped, reduction of goodwill (an asset not shown in Figure B.11), or patents and trademarks, and so on.

Depreciation

Depreciation is a noncash deduction from revenues for the reasonable exhaustion, wear and tear of, or obsolescence of, property used in the business. With respect to federal income taxes, the government has an enormous number of rules and regulations specifying how depreciation may be determined. Because these regulations change somewhat from year to year, new project evaluations should be based on the most recent regulations. Revisions in the income tax laws are often instituted with the express purpose of making capital investment more attractive by yielding a higher rate of return.

In the *straight-line (SL) depreciation*, it is assumed that the equipment value declines linearly with respect to time. The annual depreciation cost (d) is

$$d = \frac{I_0 - S_v}{n} \quad (\text{B.7})$$

where I_0 = capital investment (in dollars)
 S_v = salvage value (in dollars)
 n = economic life (years)

The book value of the equipment can be found for any year j as $I_0 - jd$. For example, if the investment $I_0 = \$10,000$ and the salvage value $S_v = \$1000$, the annual depreciation for an asset with a 5-year life is $\$9000/5 = \1800 .

Property other than buildings (18-year property) placed into service at the present time must use the *modified accelerated cost recovery system (MACRS)* in calculating depreciation. Property is classified as having 3, 5, 7, 10, 15, or 20 years life. Some examples are:

- Three-year: Special tools, televisions, furniture, computers
- Five-year: Cars, light trucks, technological equipment, telephone switching, research equipment
- Seven-year: Office furniture and fixtures
- Ten-year: Barges, fruit bearing trees

Rather than explain the complicated rationale behind the allowable rates of depreciation for those classes, Table B.5 just lists the rates for what is called accelerated depreciation (MACRS). Note that for each class you deduct some depreciation after the “useful life” (class life) expires. You can find other tables for accelerated depreciation for various circumstances in any guide to federal income taxes. If you do not want to choose accelerated rates of depreciation, you can choose straight-line depreciation (SL) using the rates listed in Table B.6. The specific

TABLE B.5
MACRS depreciation rates

(Half-year convention*)						
Year	3-year property (%)	Year	5-year property (%)	Year	7-Year property (%)	
1	33.33	1	20.00	1	14.29	
2	44.45	2	32.00	2	24.49	
3	14.81	3	19.20	3	17.49	
4	7.41	4	11.52	4	12.49	
		5	11.52	5	8.93	
		6	5.76	6	8.92	
				7	8.93	
				8	4.46	

*Half-year convention assumes the property is placed in service midyear no matter when it was actually placed in service.

TABLE B.6
Straight-line depreciation rates
(half-year convention)

Year	3-Year property (%)	5-Year property (%)	7-Year property (%)	10-Year property
1	16.6	10.00	7.14	5.00
2	33.33	20.00	14.29	10.00
3	33.33	20.00	14.29	10.00
4	16.67	20.00	14.28	10.00
5		20.00	14.29	10.00
6		10.00	14.28	10.00
7			14.29	10.00
8			7.14	10.00
9				10.00
10				10.00
11				5.00

choice of MACRS, SL, or another method is quite complex because of the extensive detailed rules for depreciation allowance and corresponding federal income tax consequences and is therefore beyond our scope here.

EXAMPLE B.4 COMPARISON OF DEPRECIATION METHODS

A piece of capital equipment costs \$6000, has a service life of 3 years, and has no salvage value. Compute the depreciation schedules using the following methods: SL and MACRS.

Solution. Assume the equipment falls into the 3-year class life schedule. The depreciation allowances are as follows:

Year	SL	MACRS
1	1000	2000
2	2000	2667
3	2000	889
4	1000	444

Salvage value

Salvage value is the price that can be actually obtained or is imputed to be obtained from the sale of used property if, at the end of its usage, the equipment (property) still has some utility. Salvage value is influenced by the current cost of equivalent equipment, its commercial value, whether the equipment must be dismantled and relocated to have utility for others, and the (projected) physical condition of the equipment. Salvage value can be thought of as a cash flow that may occur several years in the future, but does not represent income for federal income tax purposes when received.

Income taxes

The federal income tax on profits from corporations is based on income after all costs, including depreciation, have been deducted. Because depreciation affects taxable income, it is an important consideration in estimating profitability. The federal income tax rate for large corporations (profit greater than \$75,000) was recently roughly 34–35 percent. State income taxes may push the total tax rate to about 40 percent. Therefore as an expense a depreciation amount of \$1 reduces taxes about \$0.40. At this level of taxation, the before-tax rate of return will be roughly 1.67 times the after-tax rate of return.

Tax credit

Periodically Congress has permitted the use of tax credits as a direct reduction from income taxes. Examples are tax credits for installing energy conservation devices, use of alcohol fuels and electric vehicles, development of orphan drugs, creation of low-income housing, and some research expenditures. Tax credits have been used historically to stimulate capital investment in the United States. Such deductions are more valuable than depreciation because they represent direct deductions from the tax bill after taxes are computed on income.

Two other factors that need to be considered in project evaluation that are not expressly found in financial statements are inflation and debt–equity ratio.

Inflation

Inflation can be a significant factor in analysis of profitability. High inflation rates frequently occur in many countries. In computing the rate of return or net present value, you need to obtain a measure of profitability that is independent of the inflation rate. If you inflate projections of future annual income, the computed rate of return may largely result from the effects of inflation. Most companies strive for an internal rate of return (after taxes) of 10–20 percent in the absence of inflation;

this figure would rise if projected future income is increased to include the effects of inflation (i.e., selling prices are raised yearly). Furthermore, costs will also rise because of inflation.

Griest (1979) has discussed the effects of inflation on profitability analysis and has pointed out that the percentage change in profits after income taxes rarely increases at the rate of inflation, largely due to the effects of taxation. Assumptions about inflation can change the relative ranking of project alternatives based on net present value; special techniques based on probability may be required because inflation is difficult to predict.

Debt–Equity ratio

The debt-to-equity ratio quantifies the sources of funds used for capital investment and is generally expressed as percent/percent, for example, 75/25 means 75 percent debt, 25 percent equity. Debt financing involves borrowing funds (from banks, insurance companies, or other lenders, or by selling bonds) based on fixed or adjustable interest rates and specified lengths of time until the loan is due. Equity financing involves selling shares of stock or partnership shares to raise investment funds or the expenditure of retained earnings of the company. Both debt and equity financing can be used on the same project. Compared with 100-percent equity financing, the rate of return on an investment can be increased if the interest rate for borrowed capital is favorable because interest payments are considered to be an expense in computing income taxes. Suppose that the debt interest rate is 12 percent and the equity interest rate is also 12 percent. Because interest payments are deductible, the effective debt interest rate after taxes for a tax rate of 40 percent is 7.2 percent.

Next, let us go through an example of project evaluation that includes most of the factors just discussed.

EXAMPLE B.5 EVALUATION OF USING EQUITY VERSUS DEBT FINANCING

Suppose you are asked to evaluate the purchase of the multicone cyclone referred to in Example 3.4. The capital investment is \$35,000 (see Example 3.4), and the equipment has a class life of 5 years, after which it will be sold for the salvage value of \$4000. The income stream generated by the machine is on line A in Tables EB.5A and EB.5B. As the equipment ages, its operating and maintenance costs increase, and line B lists the expense profile. Assume a tax rate of 35 percent with no investment tax credit. Evaluate two possible scenarios: (a) 100 percent use of equity and (b) 100 percent debt financing. Use straight-line depreciation; for debt financing, for simplicity assume equal annual payments (principal plus interest) to the lender for the 5 years at a rate of 10.5%.

Solution. Tables EB.5A and EB.5B list the data needed in the evaluation. Depreciation is straight line (SL). The gain on sale of the cyclone at the end of the year 5 is \$500 (which is subject to ordinary income tax)

TABLE EB.5A
Calculations for purchase of cyclone (100% equity)

	Year				
	1	2	3	4	5
A. Income	\$18,000	\$28,000	\$30,000	\$30,000	\$30,000
B. Expenses	2,000	3,000	5,000	7,000	8,000
C. Profit (A - B)	16,000	25,000	25,000	23,000	22,000
D. Depreciation (straight line)	3,500	7,000	7,000	7,000	7,000
E. Net income before taxes (C - D)	12,500	18,000	18,000	16,000	15,000
F. Gain on sale at end of year 5	—	—	—	—	500
G. Income taxes (0.35(E + F))	4,375	6,300	6,300	5,600	5,425
H. Net income after taxes (E + F - G)	8,125	11,700	11,700	10,400	10,075
I. Salvage value	—	—	—	—	4,000
J. Cash flow (D + H + I)	11,625	18,700	18,700	17,400	21,075

TABLE EB.5B
100 percent debt financing for cyclone

	Year				
	1	2	3	4	5
A. Income	\$18,000	\$28,000	\$30,000	\$30,000	\$30,000
B. Expenses	2,000	3,000	5,000	7,000	8,000
Interest	3,675	3,079	2,420	1,693	889
C. Profit (A - B)	12,325	21,921	22,580	21,307	21,111
D. Depreciation (straight line)	3,500	7,000	7,000	7,000	7,000
E. Net income before taxes (C - D)	8,825	14,921	15,580	14,307	14,111
F. Gain on sale at end of year 5	—	—	—	—	500
G. Taxes (0.35(E + F))	3,089	5,222	5,453	5,007	5,114
H. Net income after taxes (E + F - G)	5,736	9,699	10,127	9,300	9,497
I. Principal payments	5,676	6,272	6,931	7,658	8,463
J. Salvage value	—	—	—	—	4,000
K. Cash flow (D + H + I + J)	3,560	10,427	10,196	8,642	12,034

Cost	\$35,000
Accumulated depreciation	<u>31,500</u>
Adjusted basis for income tax	<u>\$ 3,500</u>

Sales price	\$4,000
Basis	<u>3,500</u>
Gain	<u>\$ 500</u>

What criterion should you use to make the evaluation? You can calculate the internal rate of return for case (a) from

$$0 = \frac{-35,000}{(1+i)^0} + \frac{11,625}{(1+i)^1} + \frac{18,700}{(1+i)^2} + \frac{18,700}{(1+i)^3} + \frac{17,400}{(1+i)^4} + \frac{21,075}{(1+i)^5}$$

the solution for which is IRR = 38 percent. But what about case (b)? The input and out-flow in year 1 would be \$35,000 received as a loan, less \$35,000 paid out as the purchase price of the cyclone, leaving 0 as the initial cash flow. The IRR would be infinite!

Consequently, a better criterion for evaluation is to use the net present value for each case. Select an interest (discount) rate of 15 percent per annum.

Case (a):

$$\begin{aligned} \text{NPV}_a &= \frac{-35,000}{(1+i)^0} + \frac{11,625}{(1+i)^1} + \frac{18,700}{(1+i)^2} + \frac{18,700}{(1+i)^3} + \frac{17,400}{(1+i)^4} + \frac{21,075}{(1+i)^5} \\ &= \$52,500 \end{aligned}$$

Case (b):

$$\begin{aligned} \text{NPV}_b &= \frac{0}{(1+i)^0} + \frac{3,560}{(1+i)^1} + \frac{10,427}{(1+i)^2} + \frac{10,196}{(1+i)^3} + \frac{8,642}{(1+i)^4} + \frac{12,034}{(1+i)^5} \\ &= \$28,608 \end{aligned}$$

Clearly case (a) appears better. But other interest rates could be chosen and similar calculations made for NPV. For example, for an interest rate of 25 percent per annum

$$\text{NVP}_a = \$9,875$$

$$\text{NVP}_b = \$17,780$$

so that at the higher discount rate case (b) is preferred.

The change in the NPV using debt financing of assets is known as the principle of *leverage*. A similar result can often be obtained by leasing equipment because the lease payments are completely deductible as expenses for income tax purposes.

REFERENCES

- Aspen Technology, Inc. *Aspen Plus*. Cambridge, MA (1998).
 Brigham, E. F. *Financial Management: Theory and Practice*, 3d ed. Dryden Press, Chicago (1982).
 Brown, T. R. "Capital Cost Estimating." *Hydrocarbon Processing*, pp. 93–100 (October, 2000).
 CHEMCOST. *Process Equipment Cost Estimation*. Icarus Corp., Rockville, MD (1999).
 Garrett, D. E. *Chemical Engineering Economics*. Kluwer, New York (1989).
 Green, D. W.; and J. O. Maloney, eds. *Perry's Chemical Engineering Handbook*, 7th ed. McGraw-Hill, New York (1997).

- Griest, W. H. "Making Decisions in an Inflationary Environment." *Chem Eng Prog* p. 13 (June, 1979).
- HYSYS. Hyprotech Ltd. Calgary, Alberta, Canada (1998).
- Jelen, F. C.; and J. H. Black. *Cost and Optimization Engineering*. McGraw-Hill, New York (1983).
- Kuri, C. J.; and A. B. Corripio. "Two Computer Programs for Equipment Cost Estimation and Economic Evaluation of Chemical Processes." *Chem Engr Educ* pp. 14–17 (Winter, 1984).
- Oil and Gas Journal*. Tulsa, OK.
- Ostwald, P. F. *Engineering Cost Estimating*, 3rd ed. Prentice-Hall, Upper Saddle River, NJ (1992).
- ProII*. Simulation Sciences, Brea, CA (1998).
- Rusnak, I.; A. Guez; and I. Bar-Kana. "Multiple Objective Approach to Adaptive Control of Linear Systems." In *Proceedings of the American Control Conference*. San Francisco, June 1993, pp. 1101–1105.
- Seider, W. D.; J. D. Seader; and D. R. Lewin. *Process Design Principles*. John Wiley, New York (1999).
- Turton, R.; R. C. Baile; W. B. Whiting; and J. A. Shaeiwitz. *Analysis, Synthesis, and Design of Chemical Processes*. Prentice-Hall, Upper Saddle River, NJ (1998).
- Woinsky, S. G. "Use Simple Payout Period to Screen Projects." *Chem Eng Prog* pp. 33–37 (June, 1996).

NOMENCLATURE

a	coefficient in quadratic function
a_i	lower bound on constraint function
a_{pi}	crude oil yield
A	annual revenue
A	area
A_i^l	lower bound on analysis for component i
A_i^u	upper bound on analysis for component i
A	coefficient matrix in linear constraint
A	Jacobian matrix of constraints
\bar{A}	adjoint matrix of A
AL	augmented Lagrangian function
b	coefficient in quadratic function
b_i	i th parameter estimate
b_i	upper bound on constraint function
b_{ij}	coefficients in quadratic function
\mathbf{b}	vector of coefficients in equality constraints
B	barrier function
\mathbf{B}	basis matrix
\mathbf{B}	approximation to the Hessian matrix used in sequential quadratic programming
c	coefficient in quadratic function
c	constant in rate of convergence
c_j	right-hand side, inequality constraint
\mathbf{c}	vector of cost coefficients
C	cost
C_B	base cost

$C_{j,k}$	completion time (see Example 16.2)
d_j	depreciation taken in year j
d_{ij}	coefficient in quadratic function
d^k	vector in BFGS method
\mathbf{d}_c	search vector
D	number of units produced in manufacturing
D	diameter
D_j	cumulative depreciation in year j
D_p	maximum demand
\mathbf{D}	diagonal matrix
dn_i	negative deviation variable
dp_i	positive deviation variable
e	measurement error
e_i	i th eigenvalue
E	error measure
E	annual expense
f	objective function
f_{low}	lowest estimated value of f
F	reduced objective function
F_i	future value or payment in year i
g_j	inequality constraint
$g(\alpha)$	line search objective function
\mathbf{g}	vector of inequality constraints
h	step size in discretization
h_j	equality constraint
\mathbf{h}	vector of equality constraints
\mathbf{H}	Hessian matrix
$\tilde{\mathbf{H}}$	modified Hessian matrix in Equation (6.16)
\mathbf{H}^k	approximation of \mathbf{H} at iteration k
\mathbf{H}^{-1}	inverse Hessian matrix
i	interest rate
I_x	cost index factor
\mathbf{I}	identity matrix
J	mathematical operator
\mathbf{J}	Jacobian matrix
k	iteration number
k_i	cost coefficients
\mathbf{l}	lower bound
L	Lagrangian
L_i, l_i	lower bound
\mathbf{L}	lower triangular matrix
m	slope
m	number of equality constraints
m	rank of matrix
m	number of control moves

m_e	number of independent equality constraints
m_i	number of independent inequality constraints
M	penalty coefficient (big M method)
M_i	i th minor of matrix
\mathbf{M}	matrix in MINLP
n	dimension of \mathbf{x}
n	number of time periods in investment project
n	number of data sets
$n1$	negative deviation variables
N	number of terms in Equation (16.1)
\mathbf{N}	matrix involving nonbasic variables
p	number of variables
p	total number of constraints
p	order of convergence
p	prediction horizon
$p1$	positive deviation variables
$p^{(i)}$	job scheduling index
P	present value
P,p	penalty function
\mathbf{P}	job scheduling vector
q	weighting factor in model predictive control
Q	production level
\mathbf{Q}	weighting matrix in quadratic programming
\mathbf{Q}	positive-definite matrix
r	number of inequality constraints
r	repayment multiplier
r	line search objective
$r(k)$	setpoint
r	penalty function weighting coefficient
s_i	slack or surplus variable
s_i	component of a search direction
\mathbf{s}	search direction
S	size parameter
S_i	supply limit
S_i	step response coefficient
S_v	salvage value
S_c^{Ki}	relative sensitivity of cost to coefficient K_i
t	time
T	simulated annealing variable
$t_{j,k}$	processing time (see Example 16.2)
\mathbf{u}	vector of Lagrange multipliers
\mathbf{u}	upper bound
U_i, u_i	upper bound
$u(k)$	manipulated variable
v_i	value coefficient

v_i	i th eigenvector
v_i	vector defined in necessary conditions
V_j	book value in year j
V	optimal objective value
\mathbf{V}	eigenvector matrix
\mathbf{V}	variance–covariance matrix
w	factor in Equation (5.18)
w_i	weighting factor in Equation (16.2)
w_1	positive weight in penalty function
x_{ij}	variable in assignment problem
\mathbf{x}	vector of n variables
\mathbf{x}	model input vector
\mathbf{x}_B	vector of basic variables
\mathbf{x}_D	dependent variables
\mathbf{x}_I	independent variables
\mathbf{x}^k	optimization variable at iteration k
\mathbf{x}_N	vector of nonbasic variables
\mathbf{x}^P	reference point
\mathbf{x}_T	tear variables
\mathbf{x}^*	optimal value of \mathbf{x}
$\tilde{\mathbf{x}}^*$	approximation to \mathbf{x}^*
\mathbf{x}	data matrix
X_{ij}	binary variable in objective function
y	model output
Y	optimization variable
Y_m	measured variable value
Y	operating hours per year
Y_j	observed data point
\mathbf{y}	integer variable vector
z	distance variable
z	MINLP objective function term

GREEK SYMBOLS

α	distance moved along a search vector (step length)
β	positive weighting factor
β_j	model parameter
$\hat{\beta}_j$	estimated model parameter
β^k	step size adjustment in conjugate gradient method
γ	positive weighting factor
δ	bound on step size
δ	parameter in convexity definition
∇	gradient operator (“del”)

Δ	difference in general
Δ	determinant
Δt	discretization in time for model predictive control
Δu	change in manipulated variable
$\Delta \mathbf{x}^k$	$\mathbf{x}^{k+1} - \mathbf{x}^k$
Δ_i	determinant of i th principal minor
ε	roundoff error
ε_i	convergence (termination) criterion
ε_j	random error between j th data point and model prediction
λ	vector of Lagrange multipliers
θ	angle between two vectors
ρ	a scalar between 0 and 1
$\tau_{j,k}$	scheduling variable (see Example 16.2)
ϕ_k	vector of tear variables in flowsheet optimization

SUPERSCRIPTS

k	stage in search
T	transpose
o	at optimal solution
opt	optimum
'	first derivative
*	optimum

NAME INDEX

- Abadie, J., 306, 328
Abel, O., 514
Adjiman, C. S., 373, 412, 496, 513
Agnew, J. B., 449, 458
Aldrich, C., 413
Alkaya, D., 548
Amundson, N. R., 142, 600
Anderson, J. L., 459
Androulakis, I. A., 373, 412, 496, 498–500, 513
Angeline, P. J., 413
Aragon, C. R., 399, 412
Arlt, W., 452, 453, 458
Armijo, L., 205, 210
Ashdee, B. T., 478
Athier, G., 419, 438
Avriel, M., 142, 186, 210, 395, 411
Azzaro-Pantel, C., 413
- Backx, T., 575, 579
Badgwell, T. A., 508, 513, 568, 581
Baile, R. C., 329, 604, 610, 629
Baker, T. E., 306, 328, 553, 554, 565, 579, 580
Balakrishna, A., 548
Balakrishna, S., 514
Bar-Kana, I., 84, 104, 629
Barnes, J. W., 393, 395, 411, 412
Barsky, B., 116, 142
Bartela, R., 116, 142
Bates, D. M., 62, 73
Baudet, P., 413
Bauer, M. H., 443, 446, 458
Baumol, W. J., 20, 27
Bazarra, M. S., 142, 159, 176
Beatty, J., 116, 142
Becker, H. A., 176, 179
Beightler, C. S., 177
Bejan, A., 478
Bendor, E. A., 73
Bequette, B. W., 73, 459, 501, 503–505, 508, 513
Berna, T. J., 319, 328
Bernal-Haro, L., 413
Beveridge, G. S. G., 6, 27, 138, 142, 176
Bhaskar, V., 104
Bhatia, T. K., 329
Biegler, L. T., 329, 372, 443, 458, 459, 514, 529, 542–548, 577, 581
Biles, W. E., 66, 73
Bird, R. B., 51, 69, 73
Bischof, C., 546
Bischoff, K. B., 481, 513
Bixby, R. E., 238, 253
Black, J. H., 610, 617, 629
Blank, L. T., 105
Boddington, C. E., 581
Boggs, P. T., 211
Bonvin, D., 514
Bordons, C., 568, 580
Borwein, J., 142
Bosgra, O., 575, 579
Boston, J. F., 548
Bowman, M. S., 105
Box, G. E. P., 48, 60, 62, 66, 73
Brent, R. P., 177, 211
Brigham, E. F., 617, 618, 628
Brinn, M. S., 423, 427, 439
Briones, V., 419, 439
Britt, H. I., 548
Brooke, A., 323, 328
Brooks, S. A., 329
Brown, G. G., 290, 328
Broyden, C. G., 208, 210, 211
Brummerstedt, E. F., 88, 104
Brunger, A. T., 496, 513
Bryant, G. F., 553, 554, 565, 579
Bunch, J. R., 598, 600
Bunch, P. R., 561, 580
Burwick, C. W., 492, 513
Byrd, R. H., 211
- Caballero, J. A., 373
Camacho, E. F., 568, 580
Campbell, H. G., 600
Canada, J. R., 105
Carle, A., 546
Carpentier, J., 306, 328
Carroll, J. A., 478
Cerde, J., 373

- Chaudhuri, P. D., 439
 Chen, H. S., 543, 546, 548
 Cheng, Y., 576, 581
 Choi, H., 413
 Churchill, S. W., 73
 Cichelli, M. T., 423, 427, 439
 Ciric, A. R., 439
 Colmenares, T. R., 439
 Converse, A. O., 488–490, 513
 Cook, L. N., 176, 179
 Cooley, B., 581
 Cooper, L., 177
 Corliss, G. F., 546, 547
 Cornellisen, R. L., 439
 Corripio, A. B., 609, 629
 Coville, A. R., 492, 513
 Crainic, T. G., 390, 411
 Cramer, S. M., 459
 Crellin, R., 253
 Crowder, H. P., 373
 Crowe, C. M., 576, 580, 581
 Cugini, J. C., 254
 Currie, J. C., 478
 Curtis, A. R., 535, 547

 Daichendt, M. M., 439, 548
 Dallwig, S., 412
 Daniel, J. W., 601
 Dantzig, G. B., 223, 227, 230, 232, 239, 253
 Darst, R. B., 253
 Davidson, H., 577, 580
 Davis, J. F., 554, 565, 578, 579, 581
 Davis, M. E., 73
 de Gouvêa, M. T., 329
 Deb, K., 413
 Dell, R. F., 290, 328
 Dembo, R. S., 195, 210
 Demenech, S., 413
 Demmel, J. W., 601
 Denn, M. M., 73
 Dennis, J. E., 155, 161, 176, 187, 203, 205, 208, 210
 Dennis, J. E., Jr., 329
 Detiz, D., 39, 73
 Devanathan, S., 576, 581
 DiBella, C. W., 328, 343
 Diwekar, U. M., 439, 548
 Dixon, L. C. W., 183, 210
 Domenech, S., 399, 400, 411
 Dongarra, J. J., 598, 600
 Doty, D. R., 478
 Douglas, J. M., 508, 510, 511, 513
 Douglas, P. L., 176, 179
 Drain, D., 63, 73
 Draper, N. R., 60, 66, 73

 Dreisbach, D., 452, 458
 Drud, A., 321, 328
 Duennebier, G., 458
 Duff, I. S., 526, 547
 Duong, D. D., 74
 Duran, M. A., 369, 371, 373, 439
 Duvall, P. M., 329

 Edgar, T. F., 62, 73, 354, 373, 508, 513, 514, 566, 567, 570, 577, 578, 580, 581
 Eduljee, H. E., 454, 458
 Edwards, K., 514
 Eisenstat, S. C., 195, 210
 El-Halwagi, M., 458
 Engell, S., 413, 561, 581
 Erisman, A. M., 526, 547
 Esplugas, S., 439
 Esposito, W. R., 413
 Espura, A., 559, 580
 Evans, B., 543, 547
 Eykhoff, P., 38, 73

 Fabbri, G., 439
 Fan, L. T., 74
 Fan, Y. S., 328
 Feinberg, M., 514
 Ferguson, J. E., 419, 422, 439
 Fiacco, A. V., 525, 547
 Finlayson, B., 503, 513
 Fletcher, R., 194, 195, 208, 210
 Floquet, R., 399, 400, 411, 419, 438
 Floudas, C. A., 363, 369, 371, 373, 388, 398, 412, 413, 439, 458, 459, 496, 498, 513, 514, 548
 Fogel, D. B., 402, 412, 413
 Fogler, H. S., 481, 513
 Forbes, F., 576, 580
 Forrest, J. J., 238, 253
 Fourer, R. D., 245, 253, 323, 328
 Fox, R. L., 139, 142, 175, 176
 Fragar, E. S., 443, 458, 508, 511–513
 Freeman, M., 413
 Frey, Th., 443, 458, 466
 Friedly, J. C., 73
 Friese, T., 413
 Froment, G. F., 481, 513
 Fylstra, D. L., 322, 328, 360, 373

 Galli, M. R., 373
 Gaminibandara, K., 446, 447, 458
 Garcia, C. E., 568, 569, 580, 581
 Garrad, A., 413
 Garrett, D. E., 94, 104, 604, 605, 610, 618
 Gass, S. I., 253
 Gay, D. M., 323, 328

- Gebhart, B., 54, 73
Geddes, D., 514
Gelb, A., 577, 580
Geoffrion, A. M., 370, 373
Gill, P. E., 195, 210, 253, 517, 547
Gilliland, E. R., 140, 142
Glanz, S., 448, 458
Glover, F. A., 252, 253, 392, 393, 395, 397, 402, 408, 411, 412
Gmehling, T., 452, 453, 458
Goldfarb, D., 208, 210, 238, 253
Golub, G. H., 584, 596, 600
Gooding, W. B., 581
Gordon, S. R., 100, 104
Graves, D. B., 501, 502, 513
Greeff, D. J., 413
Green, D. W., 354, 373, 533, 535, 547, 604, 618
Griest, W. H., 618, 626
Griewank, A., 546, 547
Gross, B., 413
Grossmann, I. W., 369, 371–374, 413, 419, 439, 529, 542, 546–548, 556, 558, 580
Guez, A., 84, 104, 629
Gunderson, T., 519, 547
Guntern, C., 514
Gupta, S. K., 104, 514
- Hale, J. C., 554, 565, 579, 581
Hanagandi, V., 413
Happell, J., 88, 89, 104, 142, 148
Harding, S. T., 459
Harjunoski, I., 374
Harriott, P., 439, 465, 478
Hartland, S., 449, 458
Haupt, R. L., 413
Heinemann, R. F., 478
Helbig, A., 514
Hendon, S. R., 516, 547
Henson, M. A., 577, 578, 580
Hestenes, M. R., 211
Hext, G. R., 185, 211
Hildebrandt, D., 514
Hill, J. W., 60, 73
Hillier, F., 27
Himmelblau, D. M., 530, 547
Himsworth, F. R., 185, 211
Hiriart-Urruty, J. D., 401, 412
Hiss, G. G., 439
Hochberg, A. K., 500, 513
Holland, C. D., 443, 447, 458
Holland, J. H., 401, 412
Hooker, J., 372, 373
Horne, R. N., 478, 479
Horoka, F., 452, 458
Hrymak, A. N., 567, 580
- Hubele, N. F., 74
Hughes, R. R., 543, 547
Huh, D., 479
Hungerbühler, K., 514
Hunter, J. S., 73
Hunter, W. G., 60, 73
Hurvich, C., 84, 104
- Ierapetritou, M. G., 496, 498–500, 513
Ilias, S., 176, 179
Iribarren, O. A., 540, 547
- Jaakola, T., 328, 330
Jackson, J. E., 55, 73
Jackson, P. J., 449, 458
James, L., 183, 210
Jegede, F. O., 419, 439
Jelens, F. C., 610, 617, 629
Jensen, K. F., 501, 502, 513
Jeter, M. W., 142
Jillier, F. S., 354, 373
Johnson, E. L., 373
Johnson, J. D., 253
Johnston, D. S., 399, 412
Jordan, D. G., 88, 89, 104, 142, 148
Joulia, X., 545, 548
Jung, J. H., 413
- Kamimura, R., 84, 104
Kaplan, W., 142
Karimi, I. A., 556, 558, 560–563, 580
Karmarkar, N., 253
Karr, C. L., 413
Kearfott, R. B., 382, 412
Keller, A. H., 514
Kelley, C. T., 211
Kennedy, C. J., 478
Kernighan, B. W., 323, 328
Kim, N., 298, 301, 329
Kimpel, R. R., 328, 346
Kinnear, K. E., Jr., 413
Kisala, T. P., 548
Klein, M., 328, 346
Klepeis, J., 496, 498–500, 513
Klingman, D., 252, 253
Kluzik, H., 543, 547
Ko, J. W., 413
Koehret, B., 545, 548
Kokossis, A. C., 373, 419, 439, 514, 548
Kolari, M., 254
Kondili, E., 581
Koppel, L. B., 428, 429, 439
Kravanja, Z., 373, 548
Ku, H. M., 560–563, 580
Kubera, T., 514

- Kuehn, D. R. D., 577, 580
 Kuri, C. J., 609, 629
- Laguna, M., 392, 393, 395, 397, 402, 408,
 411, 412
 Lakshmanan, A., 329, 514
 Lang, Y. D., 543, 547, 548
 Lapidus, L., 74, 542, 547
 Laporte, G., 390, 411
 Lasdon, L., 298, 301, 306, 313, 320, 322, 328,
 329, 360, 373, 553, 577, 580
 Latour, P. R., 454, 458
 Lee, B., 374
 Lee, C. H., 413
 Lee, E. S., 374
 Lee, I-B., 413
 Lee, J. H., 479, 569, 577, 580, 581
 Lee, S., 371, 373
 Lee, Y. G., 581
 LeGoff, P., 483
 Lemarichal, C., 401, 412
 Letterman, R. D., 467, 469, 478
 Levenberg, K., 202, 210
 Levenspiel, O., 418, 513
 Lewin, D. R., 74, 517, 547, 604, 629
 Lewis, A. S., 142
 Lewis, W. K., 140, 142
 Leyffer, S., 374
 Li, J., 211
 Lieberman, G. J., 27, 354, 373
 Liebman, J. F., 320, 328
 Liebman, M. J., 577, 580
 Lightfoot, E. N., 51, 69, 73
 Lim, H. C., 428, 429, 439
 Locatelli, M., 388, 412
 Locke, M. H., 319, 328
 Logsdon, J. S., 439, 459
 Löhl, T., 413
 Lowery, R. P., 516, 547
 Luenberger, D. G., 142, 176, 271, 279, 282, 286,
 288, 291, 328, 330
 Luus, R., 328, 330, 514
 Luyben, W. L., 74
 Lyons, S. L., 478
- McAdams, W. H., 50, 140, 142, 427, 439
 McAvoy, T. J., 553, 580
 McCabe, W. L., 439, 465, 478
 Macchietto, S., 446, 458, 581
 McConville, B., 516, 547
 McCroskey, P. S., 581
 McDonald, C. M., 561, 580
 McGeoch, L. A., 399, 412
 MacGregor, J., 576, 580
 Mah, R. S. H., 540, 547, 576, 580, 581
- Mahalec, V., 543, 547
 Malone, M. F., 581
 Maloney, J. O., 604, 618
 Mangasarian, O. L., 207, 211
 Manousiouthakis, V., 413, 414, 458
 Maranas, G. D., 498, 499, 513
 Marguardt, W., 575, 579
 Marlin, T. E., 567, 576, 580
 Marquardt, W., 202, 211, 514
 Martin, G. D., 454, 458
 Martin, R. K., 223, 242, 243, 253
 Mata, J., 439
 Matias, T. R. S., 443, 458
 Mead, R., 186, 211
 Mecklenburgh, J. C., 449, 458
 Mellichamp, D. A., 62, 73, 566, 567,
 570, 581
 Meloan, C. E., 442, 458
 Meyer, D., 584, 600
 Middleman, S., 500, 513
 Miller, D. L., 582
 Mims, C. A., 481, 513
 Missen, R. W., 481, 513
 Mistree, F., 177
 Mitchell, M., 413
 Mokashi, S. D., 373
 Moler, C. B., 598, 600
 Montagna, J. M., 540, 547
 Montgomery, D. C., 62, 73, 74
 Morari, M., 373, 556, 558, 568, 569, 580, 581
 Moré, J. J., 318, 320-322, 328
 Mujtaba, I. M., 446, 458
 Murase, A., 488-490, 513
 Murray, J. E., 354, 373
 Murray, W., 195, 210, 253, 517, 547
 Murtagh, B. A., 253, 310, 321, 328
 Murty, K. G., 253
 Muske, K., 577, 578, 580
- Narasimhan, S., 413
 Nash, S. G., 142, 155, 159, 176, 195, 211, 282,
 292, 304, 305, 319, 328
 Natarajan, V., 459
 Nelder, J. A., 186, 211
 Nemethy, G., 496, 513
 Nemhauser, G. L., 353, 354, 356, 373
 Neumaier, A., 412
 Nikolaou, M., 413
 Nishikiori, N., 478
 Nocedal, J., 291, 292, 304, 328
 Noltie, C. B., 115, 142
 Novotrak, J. F., 478
- Odloak, D., 329
 Ogunnaike, T., 74

- Onken, U., 452, 453, 458
Ostwald, P. F., 604, 629
Otto, R. E., 329, 343
- Padberg, M. W., 373
Palvia, S. C., 100, 104
Pan, Y., 479
Pantelides, C. C., 458, 581, 582
Papageorgaki, S., 582
Pardalos, P. M., 496, 513
Parker, A. P., 543, 547
Parker, R. G., 374
Paules, G. E., 459
Pekny, J. F., 560, 561, 565, 580–582
Perkins, J. D., 516, 547, 582
Perry, C., 253
Peters, M. S., 27, 427, 439
Pham, Q. T., 413
Phillips, D. T., 177
Phillips, N., 252, 253
Phimister, J. R., 508, 511–513
Pho, T. K., 542, 547
Pibouleau, L., 399, 400, 411, 419, 438
Pike, R. W., 559, 580
Pilavachi, P. A., 439
Pinter, J. D., 383, 412
Pistikopoulos, E. N., 548
Poje, J. B., 254
Polley, G. T., 419, 439
Pom, R., 374
Pons, M., 545, 548
Ponton, J. W., 508, 511–513
Powell, M. J. D., 207, 211
Prett, C. E., 568, 580
Puigjaner, L., 559, 580
- Qin, J., 568, 581
Quesada, I., 548
- Ragsdell, K. M., 28, 177, 211
Rajagopalan, D., 580
Rajesh, J. K., 514
Ramagnoli, J. A., 18, 27
Raman, R., 371, 373, 548
Ramirez, W. F., 479, 526, 547
Rangaiah, G. P., 514
Rardin, R., 374
Ravindran, A., 177, 211
Rawlings, J. B., 577, 581
Ray, A. K., 104, 514
Ray, W. H., 74
Redner, R. A., 478
Reeves, C. M., 194, 195, 210
Reeves, C. R., 401, 402, 412
Reid, J. K., 526, 535, 547
- Reklaitis, G. V., 28, 177, 211, 318, 328, 459,
560, 565, 580, 582
Reppich, M., 419, 439
Rhinehart, R. R., 211
Rice, R. G., 74
Richalet, J., 568, 581
Richard, L. A., 454, 458
Riggs, J. B., 329
Rinnooy Kan, A. H. G., 388, 389, 412
Rippin, D. W. T., 554, 565, 579, 581
Ritter, K., 207, 211
Roberts, H. L., 488–490, 513
Robertson, D., 577, 581
Roenigk, K. F., 501, 513
Rollins, D. K., 576, 578, 581
Romagnoli, J. A., 575, 581
Roosen, P., 413
Rosen, J. B., 207, 211
Rosenwald, G. W., 354, 373
Rubin, E. S., 548
Rudd, D. F., 28
Rudof, R., 561, 581
Runger, G. C., 74
Rusnak, I., 84, 104, 629
Rustem, B., 290, 328, 329
- Sama, D. A., 419, 422, 439
Sanchez, M. C., 18, 27, 575, 581
Sargent, R. W. H., 446, 447, 458, 581, 582
Sarkar, S., 328
Sarma, P. V. L. N., 318, 328
Sauer, R. N., 492, 513
Saunders, M. A., 253, 310, 321, 328
Saville, B. A., 481, 513
Schechter, R. S., 6, 27, 138, 142, 176
Scheraga, H. A., 496, 513
Schittkowski, K., 211
Schlick, T., 496, 513
Schmid, C., 443, 458, 529, 543, 544, 547
Schmidt, L. D., 481, 513
Schnabel, R. B., 155, 161, 176, 187, 203, 206,
208, 210, 211, 329
Schrage, L., 254
Schrijver, A., 254, 374
Schulz, C., 413, 561, 581
Schweiger, C. A., 514
Schweyer, H. E., 439
Seader, J. D., 74, 517, 547, 604, 629
Seborg, D. E., 62, 73, 566, 567, 577, 578,
580, 581
Seider, W. D., 74, 439, 517, 547, 604, 629
Seinfeld, J. H., 74
Sen, S., 413
Setalvad, T., 501, 503–505, 508, 513
Shaewitz, J. A., 604, 610, 629

- Shaewitz, J., 329
 Shah, N., 582
 Shahbenderian, A. P., 59, 73
 Shalloway, D., 496, 513
 Shamir, R., 254
 Shanno, D. F., 208, 211
 Sherali, H. D., 142, 159, 176
 Shetly, C. M., 159, 176
 Shetty, C. M., 142
 Shobrys, D. E., 582
 Shoup, T. E., 177
 Skeel, R. D., 496, 513
 Skogestad, S., 443, 458
 Skrifvars, H., 374
 Smith, H., 60, 73
 Smith, J. C., 439, 465, 478
 Smith, S., 306, 320, 328
 Smith, W. K., 565, 581
 Snee, R. D., 254
 Sofer, A., 142, 155, 159, 176, 195, 211, 282,
 292, 304, 305, 328
 Sourander, M. L., 254
 Spendley, W., 185, 211
 Srygley, J. M., 443, 458
 Stadtherr, M. A., 543, 546, 548
 Stanley, G. M., 576, 581
 Steihang, T., 195, 210
 Steinberg, D., 177
 Steinmeyer, D. E., 419, 439
 Steur, R. E., 84, 104
 Stevens, W. F., 328, 343
 Stewart, G. W., 584, 598, 600
 Stewart, W. E., 51, 69, 73
 Stichlmair, J., 443, 446, 448, 458
 Stobbe, M., 413
 Stoecker, W. F., 176, 180
 Suh, K., 374
 Sullivan, W. G., 105
 Sung, W., 479
 Swain, J. J., 66, 73
 Swearingen, J. S., 419, 422, 439

 Tarquin, A. J., 105
 Tarrer, A. R., 428, 429, 439
 Ternet, D. J., 329
 Timmer, G. T., 388, 389, 412
 Timmerhaus, K. D., 27, 427, 439
 Tjoa, I. B., 577, 581
 Tomlin, J. A., 253
 Tong, H., 576, 581
 Trachtenberg, I., 501, 503–505, 508, 513
 Trevino-Lozano, R. A., 548
 Tsai, C. L., 84, 104
 Turkay, M., 374, 548
 Turton, R., 329, 604, 610, 629

 Uchiyama, T., 211, 220
 Ulbig, P., 413

 Van Loan, C. F., 584, 596, 600
 Vanderbei, R. J., 223, 242, 253
 Vanston, L. K., 393, 411
 Vasantharajan, S., 548
 Vasquez, M., 496, 513
 Vassiliadis, V. S., 329
 Viswanathan, J., 459

 Wahnschafft, O., 459
 Wajge, R. M., 459
 Walker, W. H., 140, 142
 Wang, K., 413
 Waren, A. D., 306, 313, 328, 360, 373
 Waton, J., 322, 328
 Watson, C. C., 28
 Watson, D. L., 561, 580
 Watson, J., 360, 373
 Watts, D. G., 62, 73
 Waven, A., 322, 328
 Weixnan, L., 177
 Wen, C. Y., 74
 Westerburg, A. W., 319, 328, 372, 459, 517, 529,
 542, 546, 547
 Westerlund, T., 374
 White, D. C., 254, 582
 White, J. A., 105
 Whiting, W. B., 604, 610, 629
 Wilde, D. J., 176, 177, 180
 Williams, T. J., 329, 343
 Williamson, C. Q., 253
 Woinsky, S. G., 617, 629
 Wolbert, D., 545, 548
 Wolsey, L. A., 243, 253, 353, 354, 356, 373
 Wood, R. K., 290, 328
 Wright, M. H., 195, 210, 253, 517, 547
 Wright, S. J., 242, 253, 291, 292, 304, 318,
 320–322, 328

 Xia, Q., 374
 Xue, G., 496, 513

 Yanniotis, S., 439
 Yeomans, H., 373
 Yocum, F. H., 516, 547

 Zagermann, S., 419, 439
 Zaher, J. J., 526, 548
 Zamora, J. M., 374, 413, 419, 439
 Zhang, J., 298, 301, 329
 Zwick, H., 514

SUBJECT INDEX

- Able to Promise, 554, 565
Active constraint, 229, 274
ADIFOR, 325
Algorithm. *See* particular method
Alkylation reactor, 492
Ammonia synthesis reactor, 488
Amortization, 623
AMPL, 323
Analytical methods. *See also* Necessary and sufficient conditions; Stationary point; Sufficient conditions
 comparison with numerical, 24, 153, 161
 constrained, 267
 computational problems, 162, 164, 175
 continuous functions, 114–151
 examples, 23, 128, 138
 n dimensions, 127, 419, 461, 464
 general conclusions, 128, 132
 one dimension, 23, 135, 161
Analytical models. *See also* Distributed system; Lumped parameter system; Simplification; Surface fitting
 definition, 43
 formulation, 44, 47
 balance equations, 39
 role of mechanisms, 41
 relationship with black box, 42, 48, 54
Applications of optimization, 9, 10, 85, 87, 89, 171, 415–548. *See also* Optimization techniques
Approximation of functions
 linear, 293
 quadratic, 197
Approximation of Hessian matrix, 208, 303
ASCEND, 519
Assignment problem, 252
Augmented Lagrangian methods, 290. *See also* Penalty function methods
Automatic differentiation, 325

Balance sheet, 618, 619, 622
Barrier function, 242, 291
Basic feasible solution in linear programming, 227
 obtaining first basic solution, 233

Basic variables, 227, 307
Basis matrix, 227, 314
Batch process optimization, 560
Batch scheduling, 559
Benders decomposition (MINLP), 370
BFGS method, 208, 304
Binary variable, 352, 407
Binding constraint, 229, 274
Black-box models
 definition, 48, 51
 response surface methods, 62
Blast furnace model, 40
Blending problem, 70
Boiler, 11, 435
Boundaries. *See also* Constraints; Kuhn-Tucker conditions; Penalty function methods; Region of search
 one dimension, 156
 effect on optimum, 168
 and optima, 119, 121, 223
 and slack variables, 226, 284
Bounds, 118, 225
Bracketing procedures, 156
Branch and bound technique, 354
 example, 355, 474
 global optimization, 385
 LP relaxation, 355
 MINLP, 361, 369
 underestimator, 385

Canonical form, 232
Capital cost estimation, 607
Capital costs, 87, 102, 604–610
Case studies for, 416–419, 514
Cash flow, 92, 102, 618, 620
Centroid, 186
Chemical plant optimization, 537
Chemical reactors, 481, 483, 492
 alkylation, 492
 ammonia, 488
 control, 571
 models, 481
 objective functions, 482
 optimal temperature, 482
 simulation, 488

- Chemical reactors—*Cont.*
 thermal cracker, 484
 tubular, 488
- Comparison of methods
 constrained, 318
 least squares, 55, 61, 577
 robustness, 318
- Complementary slackness condition, 276
- Compressor, 464
- Computer programs, sources of, 243, 319, 370, 411. *See also particular method*
- Computer-integrated manufacturing, 550
- Concave functions
 and optimization, 125
- Cone, 273, 274
- Conjugate direction. *See also Fletcher-Reeves method*
 definition, 187
 generation, 208, 209
- Conjugate gradient method, 194, 209. *See also Fletcher-Reeves method*
- CONOPT, 370
- Constrained optimization. *See also Nonlinear programming; particular technique*
 augmented Lagrangian methods, 290
 computer codes, 319 (*see also particular technique*)
 diagnosis of failures, 326
 dynamic processes, 570
 evaluation, 318
 generalized reduced gradient method, 306
 graphical illustration, 267
 Lagrange multiplier method, 278
 necessary and sufficient conditions, 281
 penalty functions, 285
 quadratic programming, 284
 successive quadratic programming, 302
 successive linear programming, 293
- Constrained optimum, 265
- Constraint qualification, 271
- Constraints. *See also Global optimization*
 active, 229, 274
 binding, 229, 274
 definition, 15, 16
 degrees of freedom, 15, 229, 294, 427, 520–523
 equality, 271, 273
 inequality, 15, 226
 linear, 223
 nonlinear, 265
 physical basis, 39, 41, 68, 70, 71
 removal, 265
- Continuous variables, 114, 116
- Contours, definition of, 132, 136
- Control, 568
- Control moves, 570
- Convergence rate. *See also specific methods*
 linear, 157
 Newton's method, 158
 order, 157
 quadratic, 157
 superlinear, 158
- Convex functions
 definition, 122
 Hessian matrix, 128
 linear case, 128
 optimization, 266, 280
- Convex programming problem, 280
- Convex region (sets), 124
- Cost estimation, 606–614
- Cost index, 612–614
- Costs
 investment capital, 93, 99, 102, 604–614
 operating, 85, 89, 102, 610–611
- Criterion of selection, 5. *See also Objective function; Profit*
- Crude oil, 556
- Cross-current extraction, 448
- Cubic interpolation, 169
- Curve fitting, 42, 51, 56, 62
- Cyclical, 540
- Cycling, 239
- Data reconciliation, 576
- Debt-equity ratio, 626
- Decision variable. *See Variable(s), independent*
- Decomposition. *See also Hessian matrix*
 complex system, 540
 generalized Benders, 370
 MINLP, 361
 need for, 19
- Degeneracy in linear programming, 239
- Degenerate vertex, 229
- Degrees of freedom, 66, 229, 294, 427, 520–523
- Dependent variable, 232, 308
- Depletion, 622
- Depreciation, 623–624
 comparison of methods, 624
 definition, 623
 MACRS, 623
 straight-line, 623
- Derivative-free methods, 183, 325. *See also Finite difference substitutes for derivatives*
- Derivatives
 approximation, 324 (*see also Finite difference substitutes for derivatives*)
 optimization, discontinuities, 115
 in simulators, 544

- Design, 516, 517
- Design costs, 606
- Design of experiments. *See* Experimental design
- Determinants, 589
 - definition, 589
- Deviation variable, 289
- Diagnosis of causes of failure of optimization codes, 326
- DICOPT, 369
- Direct methods
 - comparison of one-dimensional, 161, 163
 - conjugate search directions, 186
 - discrete-valued functions, 428
 - Nelder-Mead, 186
 - one dimension, 152–180
 - acceleration, 156
 - fixed step size, 156
 - polynomial approximation, 166
 - random search, 183
 - simplex search, 185
- Direct substitution
 - equality restrictions, 225, 265
 - slack variables, 226
- Direction of search. *See also* Conjugate direction; Steepest ascent; *specific methods*
 - conjugate, 187
 - parametric representation, 174
- Discontinuities, 114
 - effect of function, 114, 124
- Discounted cash flow, 91, 94, 100
- Discrete event dynamic systems, 565
- Discrete-valued objectives, 115, 116. *See also* Integer programming
- Disjunctive programming, 371
- Distillation (staged)
 - decision parameters, 443
 - examples, 443, 451, 453
 - extraction, 546
 - optimal design, 443
 - optimal reflux, 11, 453
- Distributed system
 - definition, 44
 - difference from lumped parameter, 45
- Eigenvalue of Hessian matrix, 128, 132, 598
- Eigenvector associated with Hessian matrix, 135, 598
- Electrostatic precipitator model, 41
- Energy conservation, 89, 102, 418
- Enterprise Resource Planning (ERP), 553
- Equality constraints. *See also* Slack variables
 - origin, 14, 38
 - SQP formulation, 302
- Equation-based optimization, 518–519, 524–525, 536
- Equations. *See* Constraints
- Equipment costs, 606
- Equipment replacement policies. *See* Depreciation
- Equity, 626
- Error in measurement, 577
- Error in numerically evaluated derivatives, 324
- Evaluation of algorithms. *See* Comparison of methods
- Evaporator, 430
- Evolutionary solver, 400
- Excel Solver
 - in LP, 245
 - in MILP, 363
 - in NLP, 322
- Expense statement, 618, 620
- Expenses, 621
- Experimental design. *See also* Factorial experimental designs
 - number of experiments, 63
 - orthogonal design, 62
- Extraction example, 448
- Extractive distillation, 546
- Extrema. *See* Optima
- Factorial experimental designs
 - two-level, orthogonal, 63
- Fathom (in branch and bound), 356
- Feasibility, 119–124, 239
- Feasible direction, 274
- Feasible points, 119–124, 239
- Feasible region, 119, 223
- Filter, 466
- Financial statements, 618
- Finite difference substitutes for derivatives, 324
 - in flowsheeting simulations, 544–545
- Fitness (in genetic algorithms), 401
- Fitting models to data, 48
- Fletcher-Reeves method
 - algorithm, 194
 - example, 196
- Flowsheet codes, 518–520
- Flowsheet optimization, 518–546
- Fluid flow examples, 461
- Full vector, 524
- Function(s)
 - approximation of (*see* Approximation of functions)
 - concave, 123, 125
 - continuous, 114
 - convex, 122, 125

- Function(s)—*Cont.*
 discontinuous, 114
 discrete, 115, 352 (*see also* Concave functions; Convex functions; Unimodal functions)
 objective, 84
 quadratic, 55, 84, 284 (*see also* Quadratic function)
 unbounded, 225
 unimodal, 155
- Future worth, 94
- GAMS, 323
- Gas compression, 464
- Gas pipeline, 469
- General polynomials
 form, 55
 n -dimensional, first degree, 55
 quadratic, 56, 587
- Generalized Benders decomposition (GBD), 370. *See also* Surface fitting, quadratic surfaces, for optimization
- Generalized reduced gradient (GRG), 306
 algorithm, 306, 527
 codes, 320
 reduced gradient, 308
 refrigeration process, 530
- Genetic algorithms, 400
- Global optimization, 382
 branch and bound, 385
 evolutionary, 400
 metaheuristics, 382
 multistart methods, 388
 scatter search, 408
 simulated annealing, 399
 tabu search, 393
- Global optimum, 121, 127, 132
- Gradient method, 189
 convergence, 192
 definition, 189
 evaluation, 207
 oscillation, 192
 reduced, 308
 steepest descent, 190
 step length, 191
- Gradient search, 189. *See also* Conjugate direction; Fletcher-Reeves method; Gradient method
- GRG, 306. *See also* Generalized reduced gradient
- GRG2, 320
- Grid search, 183
- Gross error detection, 576
- Heat exchange, 252, 418, 419–450
- Heat exchanger, 419, 422
 cost estimation, 609
 networks, 252
- Heat transfer, 418
- Hessian matrix
 approximation of, 208
 Cholesky factorization, 203
 eigenvalues of, 128, 132, 598
 eigenvectors for, 135, 598
 inverse, 202
 positive definite, 598
- Hierarchy of optimization, 6
- Hill climbing. *See* Gradient method
- Horizon, 571
- Income statement, 618, 620, 622
- Income taxes, 625
- Independent variables, 15
- Indexes, 612–614
- Inequality constraints
 form, 223
 transformation to equality constraints, 226
- Inflation, 611–614, 625
- Initial solution, 240
- Insulation thickness, 10, 89, 102
- Integer programming
 branch and bound technique, 354
 computer codes, 243, 352
- Interaction among variables, 192
- Interest rate, 94, 97, 100
- Interior point, 242–291
- Internal rate of return, 100, 102
- Inverse Hessian matrix. *See* Hessian matrix
- Investment. *See* Costs
- IRR, 615, 617. *See also* Internal rate of return
- Irreducible nets, 540
- Jacobian, 294, 598
- Job scheduling, 560
- Kalman filter, 577
- Karush-Kuhn-Tucker conditions, 267
- Knapsack problem, 352
- Kuhn-Tucker conditions, 267
- Lagrange multipliers, 271
 interpretation, 273
- Lagrangian function, 271
- Large-scale optimization, 323
- Least squares. *See also* Experimental design
 applications, 58, 61, 63, 577
 definition, 55, 61, 577
 examples, 58, 59, 451

- n dimensions, linear surface, 55
- necessary conditions, 57
- optimization technique, 57
- orthogonal design, 62
- Levels of optimization, 6
- Levenberg Marquardt method, 202. *See also specific method*
- Line search, 155, 173, 193. *See also Unidimensional search*
- Linear dependence, 593
- Linear equations, 595
- Linear independence, 593
- Linear model, 55, 223
- Linear objective function, 223
- Linear programming, 223
 - applications, 252
 - assignment problems, 252
 - basic solution, initial, 227
 - basis matrix, 227
 - canonical system, 232
 - computer codes, 243
 - degeneracy, 239
 - examples of, 245, 435
 - mixed integer programs, 243
 - modeling systems, 243
 - network flow, 252
 - phase I-phase II, 239
 - pivoting, 230
 - sensitivity analysis, 242
 - simplex method, 233
 - software, 243
 - standard LP form, 225
 - successive (in NLP), 293
 - transportation problems, 245
 - unboundedness, 238
- Linear regression. *See Least squares*
- Linearization
 - definition of linear system, 43
 - techniques, 293
- Liquid-liquid extraction. *See Extraction example*
- Lithography example, 171
- Local optima, 327, 382
- Location problem, 354
- Logarithmic barrier function, 291
- LSGRG2, 320
- Lumped parameter system
 - definition, 44
 - difference from distributed, 44
- MACRS, 623
- Manufacturing problem, 20
- Manipulated variable, 569
- Marquardt's method, 202
 - example, 203
- Material balance reconciliation, 17, 578
- Mathematical models, 37–82
- Mathematical programming, 223
- Matrix
 - basis, 227
 - condition number, 598
 - definitions, 584
 - determinant, 598
 - eigenvalues, 598
 - Hessian, 592 (*see also Hessian matrix*)
 - identity, 584
 - indefinite, 127
 - inverse, 596
 - Jacobian, 592
 - negative definite, 127
 - notation, 584
 - occurrence, 530
 - operations, 585
 - positive definite, 127, 598
 - principal minors, 589
 - rank, 594
 - semidefinite, 127
 - symmetric, 132, 584
 - transpose, 587
 - variance-covariance, 577
- Maxima. *See Optima; Sufficient conditions*
- Measurement error, 577
- Metaheuristics, 382
- Metropolis algorithm, 399
- MILP, 243, 354
- Minima. *See also Constrained optimum; Necessary and sufficient conditions; Optima; Sufficient conditions*
 - global, 118, 121, 138
 - local, 118, 123, 138
- Minimization
 - in a search direction, 133
 - unconstrained (*see Unconstrained optimization*)
- MINOS, 321
- Mixed integer linear programming, 243, 352
- Mixed integer nonlinear programming, 361
- Mixed integer programming. *See also Integer programming*
 - computer codes, 243
 - nonlinear, 361
- Model Predictive Control, 568
- Modeling systems, 322, 323
- Models. *See also Analytical models; Black-box models; Process simulators*
 - chemical reactions, 481
 - classification, 43
 - forms, 43, 48, 49
 - manufacturing, 21, 552
 - model-building, 46, 49
 - plant optimization hierarchy, 551, 553

- Modified objective function
 - Lagrangian formulation, 271
 - penalty function, 285
- Modular-based optimization, 519, 534, 537
- Module, 518, 537
- Monitoring, 575
- MPL, 323
- Multimodal functions, 135, 138
- Multistage processes, 561
- Multistart method, 388
- Mutation (in genetic algorithm), 401
- Necessary and sufficient conditions
 - first order, 128, 137
 - functions of continuous variables
 - constrained, 267
 - example applications, 269 (*see also* Kuhn-Tucker conditions; Lagrange multipliers)
 - second order, 281
 - unconstrained, 137
- Need for optimization, 4
- Negative definite, 128
- Nelder-Mead method, 186
- Nesting, 541
- Net present value, 100, 102, 615, 617
- Networks
 - generalized, 353
 - heat exchanger, 252
 - pipeline, 158, 469
- Newton-Raphson method, 197, 597
- Newton's method, 197. *See also* Quasi-Newton methods
 - advantages, 158, 161, 202
 - algorithm, 158
 - convergence, 161
 - direction of search, 197
 - disadvantages, 202
 - example, 199
 - geometric interpretation, 198
 - modified, 202
 - of solving equations, 597
 - step length, 197
- NLP (nonlinear programming) algorithm
 - advantages of different methods, 318
- NLPQL, 321
- Nodes (branch and bound), 355
- Nonlinear constraints, 265
- Nonlinear equations, solution of, 598
- Nonlinear model, 49
- Nonlinear programming problem. *See also*
 - Constrained optimization;
 - Unconstrained optimization
 - convex, 280
 - definition of, 265
 - example of, 267
 - geometric illustration, 268
- Nonlinear regression, 61, 451
- NPSOL, 321
- NPV. *See* Net present value
- Numerical evaluation of derivatives. *See* Finite difference substitutes for derivatives
- Numerical search
 - comparison, 161
 - n dimensions
 - examples, 465
 - one dimension, 152–180
 - direct methods, 166
 - examples, 161, 163, 168, 171, 431, 443, 466
 - indirect methods, 161
 - stopping criteria, 161, 168, 234, 326
- Objective. *See also* Linearization; Simulation
 - definition, 19, 84
 - economic criteria, 7, 19, 100
 - investment costs, 89, 93, 100, 604–610
 - operating costs, 85, 100, 610–611
 - profit, 100, 621, 622
- Objective function. *See also* Linearization
 - chemical reactors, 482
 - contours, 132
 - form, 131
 - linear, 223
 - simplification, 19
- Off-line optimization, 524
- Oil well location, 354
- Olefin production, 484
- One-dimensional search. *See* Numerical search; Unidimensional search
- On-line optimization, 524
- Operating cost estimation, 610
- Operating expenses, 621, 622
- Optima
 - boundaries, 119, 124 (*see also* Necessary and sufficient conditions; Stationary point)
 - conditions, 118, 126
 - existence, 118, 126
 - global, 121, 127, 132
 - local, 121, 127, 132
 - multiple extrema, 135, 382
 - n dimensions, 118
 - restricted, 121–124 (*see also* Direct substitution; Lagrange multipliers; Linear programming; Penalty function; Slack variables)
 - unrestricted, 125, 132
- Optimal control, 568
- Optimal point, 118
- Optimal scheduling, 560, 565

- Optimal solution, 14
 Optimal value. *See* Optimal solution
 Optimization
 difficulties, 26, 326
 essential features, 14
 general procedure, 19
 need for, 4
 objectives of, 4
 obstacles to, 26
 off-line, 524
 on-line, 524
 six steps, 19, 20
 strategies, 19, 265
 Optimization characteristics. *See also* Levels of optimization
 comparison of numerical and analytical, 23
 general procedure, 18
 iteration, 182
 need for optimization, 4
 objectives, 4
 opposing influences, 10, 11, 12
 Optimization complexity. *See also* Levels of optimization
 dimensionality, 19
 Optimization software, 518–520, 525
 OPTQUEST, 409
 Ordinary differential equations, 49
 Orthogonal search directions, 188
 Oscillation, 192, 299
 Outer approximation (MINLP), 369

 Parametric penalty function methods, 290
 Parametric representation, 52, 55
 Parameter estimate, 55, 58
 Payback period, 100
 Payout time, 100
 Penalty function, 285
 size of penalty, 288
 Penalty function methods
 algorithm, 285
 ill-conditioning, 286 (*see also* Lagrange multipliers)
 Penalty parameter, 285
 Penalty SLP algorithm, 298
 Phase I-Phase II procedure, 239
 Pipe diameter, 461, 469
 Planning, 553
 Plant optimization, 537
 Plant optimization hierarchy, 6, 550
 Plantwide management and optimization, 565
 Point
 feasible, 15, 118
 saddle, 127, 132, 135
 stationary, 127, 132, 135, 267

 Polynomials. *See* General polynomials; Surface fitting
 Positive definite, 128, 132, 598
 Positive-definite Hessian matrix, 128, 132, 304
 Positive-definite matrix, 128, 598
 Positive-semidefinite matrix, 128, 132
 Precedence ordering, 539
 Prediction horizon, 570
 Present value (worth), 94, 100
 Pressure vessel optimization, 87
 Problem formulation, 19
 Process design, 6, 516–517
 Process monitoring and analysis, 575
 Process operations, 551
 Process selection, 400
 Process simulators, 518–520
 Profit, 621, 622
 chemical plant, 85
 investment/profit criteria, 100
 Profitability measures, 100, 615
 Programs, computer, 243, 370
 Project evaluation, 615
 Project life, 616, 623

 Quadratic approximation, 197, 302
 geometric interpretation, 198
 Quadratic convergence, 157, 200. *See also* Conjugate direction
 Quadratic form, 132, 197
 Quadratic function
 coefficient estimation, 55, 60 (*see also* Surface fitting)
 conjugacy, 187
 geometry, 132
 minimization, 132, 187
 Quadratic interpolation, 166
 Quadratic programming, 284
 codes, 285
 quadratic programming problem, 284, 571
 Quasi-Newton methods
 algorithm, 160, 208
 BFGS, 208
 examples, 161, 163, 209
 movement in search direction, 208
 unidimensional, 160
 updating Hessian matrix, 208

 Random search methods, 183
 Rate of return. *See* Internal rate of return
 Reaction synthesis, 508
 Reactive scheduling, 553
 Reactors, chemical. *See* Chemical reactors
 Real-time optimization (RTO), 524, 565
 Recursive quadratic programming. *See* Successive quadratic programming

- Recycle systems, 509
- Reduced gradient, 308
- Reduced gradient method, generalized, 306
- Reduced vector, 524
- Reduction, 19
- Refinery application, 556
- Reflection, 187
- Reflux ratio, 443
- Refrigeration process, 530
- Region. *See* Feasible region
- Region of search, 118, 124, 274. *See also*
 - Boundaries; Constraints
 - interior optima, 118, 291
 - nonconvex, 327
- Regression. *See* Least squares
- Relative sensitivity, 25
- Repayment multiplier, 95
- Residual, 57
- Return on investment, 100
- Revenues, 614, 621
- ROI, 100
- Rosenbrock's function, 196
- Roundoff error, 324

- Saddle point
 - one dimension, 135
 - two dimensions, 127, 132
- Safeguarded Newton's method, 207
- Sales, 7
- Salvage value, 625
- Scaling, 327
 - flowsheet optimization, 526, 532
 - in one-dimensional search, 155
- Scanning (unidimensional search), 156
- Scatter search, 408
- Scheduling problem, 560
- Search methods. *See specific method*
- Secant methods. *See* Quasi-Newton methods
- Second derivatives, 127, 132, 197, 303
- Second-order-necessary conditions, 281
- Sensitivity, 25, 242, 279
- Separations processes, 441–459. *See also*
 - Distillation (staged); Extraction example
- Sequential modular flowsheeting, 524, 539
- Sequential quadratic programming. *See*
 - Successive quadratic programming
- Sequential search
 - discrete-valued objectives, 115, 116
 - one-dimensional search, 156, 168
 - simplex, 185
- Shadow price, 242, 279. *See also* Lagrange multiplier
- Simplex, 185, 233
 - Simplex method of search, 185. *See also* Linear programming
- Simplification. *See also* Decomposition; Linearization; Objective function
 - linear approximation, 19, 293
 - mathematical, 19
 - physical model, 21, 47
 - quadratic approximation, 131, 302
- Simulated annealing, 399
- Simulation, 518–520
 - sequential modular, 524, 529
 - simultaneous modular, 524, 527, 543
- Simultaneous modular model, 524, 527, 543
- Slack variables, 225
 - inequality constraints, 223
 - Lagrange multipliers, 269 (*see also* Kuhn-Tucker conditions)
 - linear systems, 230
 - sufficient conditions, 281
- Software
 - flowsheeting simulators, 544–545
 - global optimization, 411
 - LP, 223
 - MILP, 243
 - MINLP, 370
 - NLP, 319
- Spreadsheet optimizer, 243, 322
- Spreadsheets, 243, 322
- SQP. *See* Successive quadratic programming
- Stationary point, 269
 - definition, 282
 - n dimensions, 259
 - constrained, 282
 - Kuhn-Tucker conditions, 269
 - unconstrained, 127
 - one dimension, 135
 - need for higher derivatives, 135
- Steady state model, 44
- Steam generator, 435
- Steam system, 435
- Steepest ascent. *See* Gradient method
- Steepest descent. *See* Gradient method
- Step response, 570
- Step response model, 570
- Step size in search, 156, 158, 160, 190, 304, 311. *See also specific method*
- Stopping criteria. *See* Termination
- Strictly concave, 123
- Strictly convex, 123
- Suboptimization, 8
- Successive linear programming, 293
- Successive quadratic programming
 - algorithm, 302

- codes, 321
- examples, 305
- Sufficient conditions, 281. *See* Necessary and sufficient conditions
- Sum of infeasibilities, 240, 315
- Superbasic variables, 310
- Superposition, 43
- Supply chain management, 550
- Supply limit, 556
- Surface fitting, 54, 62
 - definition, 55
 - "fit", 56. *See* Least squares quadratic surfaces, 55
 - for optimization, 59
- Synthesis, 516

- Tabu search, 393
- Tax credit, 625
- Taxes, 625
- Taylor series, 136
- Tearing, 540, 541
- Termination, 194, 305, 325. *See also*
 - Convergence rate
- Thermal cracker, 484
- Time value of money, 94
- Transformation method. *See* Penalty function
- Transportation problem, 245
- Traveling salesman problem, 353
- Trust region, 206, 298
- Two-level experiment design, 184. *See* Factorial experimental designs

- Unconstrained optimization, 183

- examples, 204, 209, 451, 464
- Underestimator, 385
- Unidimensional search
 - indirect, 155, 157
 - interpolation, 166
 - multidimensional search, 173
 - polynomial approximation, 166
 - scanning and bracketing procedures, 156
- Unimodal functions
 - definition, 156
 - in numerical search, 156
- Unit management and control, 567
- Univariate search, 185
- Unsteady state model, 44, 569
- Upper bounded variables, 225, 299

- Vapor-liquid equilibrium, 451
- Variable(s)
 - basic, 227
 - continuous *versus* discrete, 352
 - dependent, 232, 308
 - independent, 232, 308
 - interaction among, 131
 - nonbasic, 232, 308
 - slack, 226, 284
- Variance-covariance matrix, 577
- Vector, 524, 584
- Vertex, linear equalities, 229

- Waste heat recovery, 419
- Weighting factor, 571

- Yield matrix, 484