Hamdy A. Taha

Operations Research
An Introduction

Eleventh Edition

50TH ANNIVERSARY

With Introduction to Analytics, Machine Learning, and Artificial Intelligence

Operations Research: An Introduction

This page intentionally left blank

Operations Research: An Introduction

Eleventh Edition

Hamdy A. Taha
University of Arkansas, Fayetteville

Pearson

Content Management: Dawn Murrin, Erin Ault
Content Production: Devanshi Tyagi
Product Management: Holly Stark
Product Marketing: Krista Clark
Rights and Permissions: Rimpy Sharma

Please contact https://support.pearson.com/getsupport/s/ with any queries on this content.

Cover Image: Maksym Ivasiuk/Shutterstock.

Library of Congress Cataloging-in-Publication Data

Names: Taha, Hamdy A., author.
Title: Operations research: an introducton / Hamdy A. Taha, University of Arkansas, Fayetteville.
Description: Eleventh edition. | Hoboken, NJ : Pearson, [2023] | Includes bibliographical references and index.
Identifiers: LCCN 2021051103 | ISBN 9780137625864
Subjects: LCSH: Operations research. | Programming (Mathematics)
Classification: LCC T57.6 .T3 2023 | DDC 658.4/034—dc23/eng/20211118
LC record available at https://lccn.loc.gov/2021051103

ScoutAutomatedPrintCode

Pearson

Rental:
ISBN-10: 0-13-762586-3
ISBN-13: 978-0-13-762586-4

Pearson's Commitment to Diversity, Equity, and Inclusion

Pearson is dedicated to creating bias-free content that reflects the diversity, depth, and breadth of all learners' lived experiences.

We embrace the many dimensions of diversity, including but not limited to race, ethnicity, gender, sex, sexual orientation, socioeconomic status, ability, age, and religious or political beliefs.

Education is a powerful force for equity and change in our world. It has the potential to deliver opportunities that improve lives and enable economic mobility. As we work with authors to create content for every product and service, we acknowledge our responsibility to demonstrate inclusivity and incorporate diverse scholarship so that everyone can achieve their potential through learning. As the world's leading learning company, we have a duty to help drive change and live up to our purpose to help more people create a better life for themselves and to create a better world.

Our ambition is to purposefully contribute to a world where:

- Everyone has an equitable and lifelong opportunity to succeed through learning.
- Our educational content accurately reflects the histories and lived experiences of the learners we serve.
- Our educational products and services are inclusive and represent the rich diversity of learners.
- Our educational content prompts deeper discussions with students and motivates them to expand their own learning (and worldview).

Accessibility

We are also committed to providing products that are fully accessible to all learners. As per Pearson's guidelines for accessible educational Web media, we test and retest the capabilities of our products against the highest standards for every release, following the WCAG guidelines in developing new products for copyright year 2022 and beyond.

You can learn more about Pearson's commitment to accessibility at **https://www.pearson.com/us/accessibility.html**

Pearson

Contact Us

While we work hard to present unbiased, fully accessible content, we want to hear from you about any concerns or needs with this Pearson product so that we can investigate and address them.

Please contact us with concerns about any potential bias at **https://www.pearson.com/report-bias.html**

For accessibility-related issues, such as using assistive technology with Pearson products, alternative text requests, or accessibility documentation, email the Pearson Disability Support team at **disability.support@pearson.com**

This page intentionally left blank

In Loving Memory of Karen

Los ríos no llevan agua,
el sol las fuentes secó . . .
¡Yo sé dónde hay una fuente
que no ha de secar el sol!
La fuente que no se agota
es mi propio corazón . . .

—V. Ruiz Aguilera (1862)

The rivers bear no water,
the sun dried up the springs . . .
I know of a source that the
sun cannot deplete!
The source that springs
forever is the beating
of my heart . . .

This page intentionally left blank

Contents

What's New in the Eleventh Edition xxv
Acknowledgments xxix
About the Author xxxi
Trademarks xxxiii

Chapter 1 Overview of OR, Analytics, AI, and ML in Decision-Making 1

1.1 Introduction *1*
1.2 Two Distinct Approaches for Making Decisions *2*
1.3 OR Mathematical Modeling *3*
1.3.1 Elements of a Mathematical Model *3*
1.3.2 Solving the Mathematical Model *6*
1.3.3 Proxy Optimization—Queuing, Simulation, and Monte Carlo Models *8*
1.3.4 OR Is More Than Just Mathematics *8*
1.4 Analytics Modeling *11*
1.4.1 Elements of the Analytics Model *11*
1.4.2 Data Relevance in Analytics *12*
1.4.3 Example Applications *13*
1.5 Artificial Intelligence (AI) *17*
1.6 Machine Learning (ML) *22*
Bibliography *24*
Problems *24*

Chapter 2 Modeling with Linear Programming 29

2.1 Two-Variable LP Model *29*
2.2 Graphical LP Solution *31*
2.2.1 Solution of a Maximization Model *32*
2.2.2 Solution of a Minimization Model *34*
2.3 Computer Solution with Solver and AMPL *36*
2.3.1 LP Solution with Excel Solver *37*
2.3.2 LP Solution with AMPL *40*
2.4 Linear Programming Applications *43*
2.4.1 Investment *44*
2.4.2 Production Planning and Inventory Control *46*

2.4.3 Workforce Planning *51*
2.4.4 City Development Planning *54*
2.4.5 Blending and Refining *56*
2.4.6 Additional LP Applications *60*
Bibliography ***60***
Problems ***60***

Chapter 3 The Simplex Method and Sensitivity Analysis 83

3.1 LP Model in Equation Form ***83***
3.2 Transition from Graphical to Algebraic Solution ***84***
3.3 The Simplex Method ***87***
3.3.1 Iterative Nature of the Simplex Method *87*
3.3.2 Computational Details of the Simplex Algorithm *89*
3.3.3 Summary of the Simplex Algorithm *95*
3.4 Starting Solution for "Ill-Behaved" LPs ***96***
3.4.1 *M*-Method *96*
3.4.2 Two-Phase Method *99*
3.4.3 New Two-Phase Method with No Artificial Variables *101*
3.5 Special Cases in the Simplex Method ***102***
3.5.1 Degeneracy *102*
3.5.2 Alternative Optima *104*
3.5.3 Unbounded Solution *106*
3.5.4 Infeasible Solution *107*
3.6 Sensitivity Analysis ***108***
3.6.1 Graphical Sensitivity Analysis—RHS and Objective Coefficients *109*
3.6.2 Algebraic Sensitivity Analysis—RHS Coefficients *113*
3.6.3 Algebraic Sensitivity Analysis—Objective Coefficients *117*
3.6.4 Sensitivity Analysis with TORA, Solver, and AMPL *122*
3.6.5 The 100% Rule of LP Sensitivity Analysis *124*
3.7 Computational Issues in Linear Programming ***129***
Bibliography ***132***
Problems ***133***

Chapter 4 Duality and Post-Optimal Analysis 157

4.1 Definition of the Dual Problem ***157***
4.2 Primal–Dual Relationships ***160***
4.2.1 Review of Simple Matrix Operations *160*

4.2.2 Simplex Tableau Layout *161*
4.2.3 Optimal Dual Solution *162*
4.2.4 Simplex Tableau Computations *165*

4.3 Economic Interpretation of Duality *167*
4.3.1 Economic Interpretation of Dual Variables *167*
4.3.2 Economic Interpretation of Dual Constraints *168*

4.4 Additional Simplex Algorithms *170*
4.4.1 Dual Simplex Algorithm *170*
4.4.2 Generalized Simplex Algorithm *172*

4.5 Post-Optimal Analysis *174*
4.5.1 Post-optimal Diagnostics *175*
4.5.2 Changes Affecting Feasibility *176*
4.5.3 Changes Affecting Optimality *179*
4.5.4 Concurrent Changes in Feasibility and Optimality *181*

4.6 Transition From Textbook to Commercial Software Treatment of Sensitivity Analysis *183*

Bibliography *184*

Problems *184*

Chapter 5 Transportation Model and Its Variants 199

5.1 Definition of the Transportation Model *199*

5.2 Nontraditional Applications of the Transportation Model *203*

5.3 The Transportation Algorithm *206*
5.3.1 Determination of the Starting Solution *207*
5.3.2 Iterative Computations of the Transportation Algorithm *211*
5.3.3 Simplex Method Explanation of the Method of Multipliers *216*

5.4 The Assignment Model *218*
5.4.1 The Hungarian Method *218*
5.4.2 Simplex Method Explanation of the Hungarian Method *221*

5.5 The Transshipment Model *222*

Bibliography *224*

Problems *224*

Chapter 6 Network Model 239

6.1 Scope and Definition of Network Models *239*

6.2 Minimal Spanning Tree Algorithm *242*

6.3 Shortest-Route Problem ***244***
6.3.1 Examples of the Shortest-Route Applications *245*
6.3.2 Shortest-Route Algorithms *247*
6.3.3 Linear Programming Formulation of the Shortest-Route Problem *253*
6.4 Maximal Flow Model ***257***
6.4.1 Enumeration of Cuts *258*
6.4.2 Maximal Flow Algorithm *258*
6.4.3 Linear Programming Formulation of Maximal Flow Model *264*
6.5 CPM and PERT ***265***
6.5.1 Network Representation *266*
6.5.2 CPM Computations *268*
6.5.3 Construction of the Time Schedule *271*
6.5.4 Linear Programming Formulation of CPM *274*
6.5.5 PERT Networks *275*
6.6 Minimum-Cost Capacitated Flow Problem ***277***
6.6.1 Network Representation *277*
6.6.2 Linear Programming Formulation *278*
Bibliography ***282***
Problems ***283***

Chapter 7 Advanced Linear Programming 301

7.1 Simplex Method Fundamentals ***301***
7.1.1 Transition from Geometric Extreme Points to Algebraic Basic Solutions *302*
7.1.2 Simplex Tableau in Matrix Form *305*
7.2 Revised Simplex Method ***307***
7.2.1 Development of the Optimality and Feasibility Conditions *307*
7.2.2 Revised Simplex Algorithm *308*
7.2.3 Computational Issues in the Revised Simplex Method *311*
7.3 Bounded-Variables Algorithm ***313***
7.4 Duality ***317***
7.4.1 Matrix Definition of the Dual Problem *317*
7.4.2 Optimal Dual Solution *318*
7.5 Parametric Linear Programming ***320***
7.5.1 Parametric Changes in C *320*
7.5.2 Parametric Changes in b *322*

7.6 Goal Programming (GP) ***324***

7.6.1 A GP Formulation *324*

7.6.2 GP Algorithms *326*

7.7 Decomposition Algorithm ***332***

7.8 Karmarkar Interior-Point Method ***340***

7.8.1 Basic Idea of the Interior-Point Algorithm *341*

7.8.2 Interior-Point Algorithm *342*

Bibliography ***351***

Problems ***351***

Chapter 8 Stochastic Linear Programming 369

8.1 Deterministic Nature of LP Modeling ***369***

8.2 Case against Associating LP Sensitivity Analysis Ranges with LP Uncertainty ***370***

8.3 Uncertainty Representation in LP Models ***372***

8.4 Solution Models for Special LPs Under Uncertainty ***374***

8.4.1 LP Instance Using Parameter Expected Value *374*

8.4.2 Two-Stage LP Instance with Expected Value of the Objective Function *375*

8.4.3 Chance-constraint LP Instance *379*

8.4.4 Robust (Infeasibility-immune) LP Instance *382*

8.4.5 Relationship Between Expected Value, Chance Constraint, and Robust LP Instances *384*

Bibliography ***385***

Problems ***385***

Chapter 9 Integer Linear Programming 389

9.1 Illustrative Applications ***389***

9.1.1 Capital Budgeting *390*

9.1.2 Set-Covering Problem *391*

9.1.3 Fixed-Charge Problem *392*

9.1.4 Either-Or Constraints *394*

9.2 Integer Programming Algorithms ***396***

9.2.1 Branch-and-Bound (B&B) Algorithm *397*

9.2.2 Cutting-Plane Algorithm *403*

9.2.3 Benders' Decomposition Algorithm *408*

Bibliography ***414***

Problems ***414***

Chapter 10 Heuristic Programming 433

10.1 Introduction *433*
10.2 Greedy (Local Search) Heuristics *434*
10.2.1 Discrete Variable Heuristic *435*
10.2.2 Continuous Variable Heuristic *437*
10.3 Metaheuristic *440*
10.3.1 Tabu Search Algorithm *440*
10.3.2 Simulated Annealing Algorithm *444*
10.3.3 Genetic Algorithm *447*
10.4 Application of Metaheuristics to Integer Linear Programs *452*
10.4.1 ILP Tabu Algorithm *453*
10.4.2 ILP Simulated Annealing Algorithm *454*
10.4.3 ILP Genetic Algorithm *456*
10.5 Introduction to Constraint Programming (CP) *459*
Bibliography *461*
Problems *461*

Chapter 11 Traveling Salesperson Problem (TSP) 471

11.1 Scope of the TSP *471*
11.2 TSP Mathematical Model *473*
11.3 Exact TSP Algorithms *477*
11.3.1 B&B Algorithm *478*
11.3.2 Cutting-Plane Algorithm *480*
11.4 Local Search Heuristics *482*
11.4.1 Nearest-Neighbor Heuristic *482*
11.4.2 Reversal Heuristic *483*
11.5 Metaheuristics *488*
11.5.1 TSP Tabu Algorithm *488*
11.5.2 TSP Simulated Annealing Algorithm *492*
11.5.3 TSP Genetic Algorithm *494*
Bibliography *498*
Problems *499*

Chapter 12 Dynamic Programming 511

12.1 Recursive Nature of Dynamic Programming (DP) Computations *511*
12.1.1 Recursive Equation and Principle of Optimality *514*
12.2 Forward and Backward Recursion *515*

12.3 Deterministic DP Applications ***517***
12.3.1 Knapsack/Fly-Away Kit/Cargo-Loading Model *517*
12.3.2 Workforce Size Model *522*
12.3.3 Equipment Replacement Model *525*
12.3.4 Investment Model *528*
12.3.5 Inventory Models *531*
12.4 Problem of Dimensionality ***531***
12.5 Probabilistic DP Applications ***533***
12.5.1 A Game of Chance *533*
12.5.2 Investment Problem *535*
12.5.3 Maximization of the Event of Achieving a Goal *538*
12.5.4 Inventory Models *541*
Bibliography ***541***
Problems ***541***

Chapter 13 Inventory Modeling 549

13.1 The Inventory Problem ***549***
13.1.1 Elements of the Inventory Optimization Model *549*
13.1.2 Role of Demand in the Development of Inventory Models *550*
13.1.3 Issues in Inventory Modeling *552*
13.2 A Supply Chain Perspective of the Inventory Problem ***553***
13.3 Deterministic Static EOQ Models ***555***
13.3.1 Classical EOQ Model *555*
13.3.2 EOQ with Price Breaks *560*
13.3.3 Multi-Item EOQ with Storage Limitation *563*
13.4 Deterministic Dynamic EOQ Models ***566***
13.4.1 No-Setup EOQ Model *567*
13.4.2 Setup EOQ DP Models *570*
13.4.3 Silver-Meal EOQ Heuristic Model *576*
13.5 Probabilistic Continuous Review Models ***579***
13.5.1 "Probabilitized" EOQ Model *579*
13.5.2 Probabilistic EOQ Model *581*
13.6 Probabilistic Single-Period Models ***585***
13.6.1 No-Setup Model (Newsvendor Model) *586*
13.6.2 Setup Model (*s-S* Policy) *588*
13.7 Probabilistic Multiperiod Model ***590***
Bibliography ***592***
Problems ***593***

Chapter 14 Yield Management 603

14.1 What is Yield Management (YM)? *603*
14.2 Nature of Demand versus Price in YM *605*
14.2.1 Fare Segmentation *606*
14.2.2 Booking Limits Versus Protection Levels *607*
14.2.3 Optimum Protection Level in a Two-Fare Situation *610*
14.2.4 Overbooking *613*
14.3 Elasticity and Fare Pricing *615*
14.4 Concluding Remarks *620*
Bibliography *620*
Problems *620*

Chapter 15 Decision Analysis and Games 625

15.1 Special Decision-Making Situations *625*
15.2 Decision-Making Under Certainty—Analytic Hierarchy Process (AHP) *626*
15.3 Bayes' Probabilities with ML Applications *633*
15.3.1 Bayes-Based Medical Diagnostics *641*
15.3.2 Bayes-Based Machine Learning (ML) Algorithms *642*
15.4 Decision Tree Models *649*
15.4.1 Prior Probability Decision Trees *649*
15.4.2 Bayesian Decision Trees *650*
15.5 Decisions Under Uncertainty *655*
15.6 Game Theory *658*
15.6.1 Optimal Solution of Two-Person Zero-Sum Games *659*
15.6.2 Solution of Mixed Strategy Games *661*
Bibliography *666*
Problems *667*

Chapter 16 Markov Chains 685

16.1 Definition of a Markov Chain *685*
Markov process. *686*
16.2 Absolute and *n*-Step Transition Probabilities *688*
16.3 Classification of the States in a Markov Chain *689*
16.4 Ergodic (Regular) Markov Chain *690*
16.4.1 Steady-State Probabilities *691*
16.4.2 Mean First-Return (or Recurrence) Time *693*
16.5 First Passage Time *694*

16.6 Analysis of Absorbing States *697*
Bibliography *700*
Problems *700*

Chapter 17 Markovian Decision Process 711

17.1 Scope of the Markovian Decision Problem *711*
17.2 Finite-Stage Dynamic Programming Model *713*
17.3 Infinite-Stage Model *715*
17.3.1 Exhaustive Enumeration Method *715*
17.3.2 Policy Iteration Method without Discounting *718*
17.3.3 Policy Iteration Method with Discounting *721*
17.4 Linear Programming Solution *723*
Bibliography *727*
Problems *727*

Chapter 18 Queuing Systems 731

18.1 Why Study Queues? *731*
18.2 Elements of a Queuing Model *733*
18.3 Role of Exponential Distribution *734*
18.4 Pure Birth and Death Models (Relationship between the Exponential and Poisson Distributions) *735*
18.4.1 Pure Birth Model *735*
18.4.2 Pure Death Model *738*
18.5 General Poisson Queuing Model *740*
18.6 Specialized Poisson Queues *743*
18.6.1 Steady-State Measures of Performance *745*
18.6.2 Single-Server Models *747*
18.6.3 Multi-Server Models *752*
18.6.4 Machine Service Model—$(M/M/R):(GD/K/K)$, $R < K$ *757*
18.7 $(M/G/1):(GD/\infty/\infty)$—Pollaczek–Khintchine (P-K) Formula *759*
18.8 Other Queuing Models *760*
18.9 Queuing Decision Models *761*
18.9.1 Cost Models *761*
18.9.2 Aspiration Level Model *764*
Bibliography *765*
Problems *766*

Chapter 19 Discrete-Event and Monte Carlo Simulations *785*

19.1 What is Simulation? *786*
19.1.1 Discrete-Event Simulation *786*
19.1.2 Simulation Clock *788*
19.1.3 Measures of Performance of Simulated Systems *790*
19.2 Monte Carlo Simulation *792*
19.2.1 Sampling Methods *794*
19.2.2 Monte Carlo Applications *799*
19.3 Spreadsheet-Based Simulation of a Single-Server Model *806*
19.4 Methods for Gathering Statistical Observations *807*
Bibliography *811*
Problems *812*

Chapter 20 Classical Optimization Theory *819*

20.1 Unconstrained Problems *819*
20.1.1 Necessary and Sufficient Conditions *820*
20.1.2 The Newton–Raphson Method *822*
20.2 Constrained Problems *825*
20.2.1 Equality Constraints *825*
20.2.2 Inequality Constraints—Karush-Kuhn-Tucker (KKT) Conditions *833*
Bibliography *837*
Problems *837*

Chapter 21 Nonlinear Programming Algorithms *841*

21.1 Unconstrained Algorithms *841*
21.1.1 Direct Search Methods *841*
21.1.2 Gradient Method *848*
21.2 Constrained Algorithms *850*
21.2.1 Separable Programming *851*
21.2.2 Quadratic Programming *858*
21.2.3 Chance-Constrained Programming *862*
21.2.4 Linear Combinations Method *866*
21.2.5 SUMT Algorithm *868*
Bibliography *869*
Problems *869*

Chapter 22 Case Analysis 873

Case 1: Crowdsourcing Analytics and Operations Research Expertise . . . Tales of Two Studies with Different Outcomes *874*
Case 2: Airline Fuel Allocation Using Optimum Tankering *877*
Case 3: Optimization of Heart Valves Production *884*
Case 4: Scheduling Appointments at Australian Tourist Commission Trade Events *888*
Case 5: Saving Federal Travel Dollars *893*
Case 6: Optimal Ship Routing and Personnel Assignments for Recruits to the Royal Thai Navy *896*
Case 7: Allocation of Operating Room Time in Mount Sinai Hospital *904*
Case 8: Optimizing Trailer Payloads at PFG Building Glass *908*
Case 9: Optimization of Crosscutting and Log Allocation at Weyerhaeuser *916*
Case 10: Layout Planning for a Computer Integrated Manufacturing (CIM) Facility *920*
Case 11: Booking Limits in Hotel Reservations *928*
Case 12: Casey's Problem: Interpreting and Evaluating a New Test *931*
Case 13: Ordering Golfers on the Final Day of Ryder Cup Matches *934*
Case 14: Kroger Improves Pharmacy Inventory Management *936*
Case 15: Inventory Decisions in Dell's Supply Chain *941*
Case 16: Forest Cover Change Prediction Using Markov Chain Model: A Case Study on Sub-Himalayan Town Gangtok, India *944*
Case 17: Analysis of an Internal Transport System in a Manufacturing Plant *947*
Case 18: Telephone Sales Workforce Planning at Qantas Airways *949*

Appendix A Statistical Tables 957

Appendix B Partial Answers to Selected Problems 961

Appendix C AMPL Modeling Language 1003

C.1 Rudimentary AMPL Model *1003*
C.2 Components of AMPL Model *1004*

C.3 Mathematical Expressions and Computed Parameters *1011*
C.4 Subsets and Indexed Sets *1013*
C.5 Accessing External Files *1015*
C.5.1 Simple read files *1015*
C.5.2 Using `print` or `prinf` to retrieve output *1016*
C.5.3 Input table files *1017*
C.5.4 Output table files *1020*
C.5.5 Spreadsheet input/output tables *1021*
C.6 Interactive Commands *1022*
C.7 Iterative and Conditional Execution of AMPL Commands *1024*
C.8 Sensitivity Analysis Using AMPL *1025*
C.9 Selected AMPL Models *1026*
Bibliography *1039*
Problems *1039*

Appendix D Review of Vectors and Matrices 1043

D.1 Vectors *1043*
D.1.1 Definition of a Vector *1043*
D.1.2 Addition (Subtraction) of Vectors *1043*
D.1.3 Multiplication of Vectors by Scalars *1044*
D.1.4 Linearly Independent Vectors *1044*
D.2 Matrices *1044*
D.2.1 Definition of a Matrix *1044*
D.2.2 Types of Matrices *1045*
D.2.3 Matrix Arithmetic Operations *1045*
D.2.4 Determinant of a Square Matrix *1047*
D.2.5 Nonsingular Matrix *1049*
D.2.6 Inverse of a Nonsingular Matrix *1049*
D.2.7 Methods of Computing the Inverse of a Matrix *1050*
D.2.8 Matrix Manipulations Using Excel *1058*
D.3 Quadratic Forms *1059*
D.4 Convex and Concave Functions *1060*
Bibliography *1060*
Problems *1061*

Appendix E Review of Basic Probability 1063

E.1 Laws of Probability *1063*
E.1.1 Addition Law of Probability *1064*
E.1.2 Conditional Law of Probability *1065*

E.2 **Random Variables and Probability Distributions** *1065*
E.3 **Expectation of a Random Variable** *1067*
E.3.1 Mean and Variance (Standard Deviation) of a Random Variable *1068*
E.3.2 Joint Random Variables *1069*
E.4 **Four Common Probability Distributions** *1071*
E.4.1 Binomial Distribution *1072*
E.4.2 Poisson Distribution *1072*
E.4.3 Negative Exponential Distribution *1073*
E.4.4 Normal Distribution *1074*
E.5 **Empirical Distributions** *1076*
Bibliography *1081*
Problems *1081*

Appendix F Forecasting Models 1087

F.1 **Moving Average Technique** *1087*
F.2 **Exponential Smoothing** *1089*
F.3 **Regression** *1090*
Bibliography *1094*
Problems *1094*

Index 1097

This page intentionally left blank

List of Aha! Moments

Chapter 1: Nature's Ultimate Optimizers, or How Bees Construct Honeycombs! *6*

First-Ever Algorithm Programmer—Ada Lovelace. *7*

The Value of Data . . . or How a Car Dealership Embraced "Analytics" Long before the PC Era! *11*

Feeding the Dispossessed . . . or How World Food Programme (WFP) is Harnessing Analytics, OR, and Data Mining to Eradicate World Hunger! *16*

AI to the Rescue . . . or How Stephen Hawking Regained Voice! *20*

Chapter 3: The Birth of Optimization, or How George B. Danzig Developed the Simplex Method. *89*

Chapter 5: A Brief History of the Transportation Model. *203*

By Whatever Name, NW Rule Boasts Elegant Simplicity! *211*

Chapter 6: It Is Said that a Picture Is Worth a Thousand Words! *242*

In "Praise" of Google Maps, or How I Got Lost Trying to Reach a Pre-Wedding Picnic! *257*

Chapter 7: Early-On Implementations of the Simplex Algorithm, or How the Product Form of the Inverse Came About *313*

Satisficing versus Maximizing, or How Long to Age Wine! *328*

Chapter 9: Seminal Development of Dantzig-Fulkerson-Johnson Cut. *408*

Chapter 10: Earliest Decision-Making Heuristic—The Franklin Rule. *434*

Chapter 11: Earliest Mathematical Model in Archaeology, or How TSP "Seriates" Ancient Egyptian Graves *472*

TSP Computational Experience—Culminating with Leonardo da Vinci's *Mona Lisa* Reproduction *487*

Chapter 12: Solving the Marriage Problem . . . with Dynamic Programming! *514*

Chapter 13: EOQ History, or Giving Credit where Credit Is Due! *559*

Chapter 14: The Rise and Fall of People Express . . . or How American Airlines Pioneered Yield Management. *604*

Chapter 15: Mark Twain Gives "Statistics" a Bum Rap! *633*

Win or Lose? . . . or How the Monty Hall Problem Caused a "Bayesian" Stir in the US! *637*

The Futile Two-Year Search for Lost Flight AF447 . . . and How Bayesian Inference Saved the Day! *639*

An Eighteenth-Century Lottery that Yields Infinite Expected Payoff, or Does It? *653*

Cooperation Should Be the Name of the Game! *663*

Chapter 16: Spammers Go Markovian! *687*

Chapter 18: Perception of Waiting, and the Cultural Factor! *733*

The Last Will Be First . . . , or How to Move Queues More Rapidly! *744*

Chapter 19: Are We Living in a Computer Simulation? *785*

Birth of the Monte Carlo Method . . . or How the Atomic Bomb was developed! *794*

Chapter 21: Fibonacci Sequence Golden Ratio, φ, and the Quest for Intelligent Design! *846*

Appendix E: Teaching (Probability) by Example: The Birthday Challenge! *1064*

What's New in the Eleventh Edition

In previous editions, concern about the size of the hard copy prompted the inconvenient practice of placing a sizeable chunk of text material in a companion website. In the new era of e-publishing, such concern is not a factor. The new eleventh e-edition, comprised of 22 chapters and 6 appendixes, avails in one location all the material from tenth edition and its companion website, plus a significant amount of new material. The digital format should facilitate future on-demand updating of the book.

The first ten editions of *Operations Research: An Introduction* concentrated on classical Operations Research (OR) algorithms. Analytics, Artificial Intelligence (AI), and Machine Learning (ML)—the newcomers on the decision-making scene—vie OR in importance, in the sense that they all seek sound decisions, albeit from different perspectives: OR employs (mostly) mathematical models in search of the optimum (or near optimum) solution. The new fields are driven by the analysis of data for the purpose of uncovering unknown trends and relationships that will strengthen and streamline the decision-making process. Because AI and ML applications rely strongly on OR-style optimization, it is natural and timely that such applications be introduced in traditional OR courses.

Coincidental with my decision to introduce the new topics in the eleventh edition was the announcement in December 2018 by the Institute for Operation Research and the Management Science (INFORMS) to change the name of its applications-oriented flagship journal, *Interfaces*, to *Journal of Applied Analytics* "to reach a wider audience . . . [of] professionals who identify with a journal whose title includes [the word Analytics]."

This edition maintains the time-proven pedagogical features of the first ten editions:

- All algorithmic details are explained by carefully chosen numerical examples that contribute to one's intuition regarding the general problem. Theorems and proofs are used only when needed to maintain continuity.
- The focal points that unify algorithms *within* an optimization area (e.g., LP) are stressed to provide insight about the functionality of each algorithm. For example, the plethora of available simplex method algorithms may give the impression that they are fundamentally different when, in fact, they all are based on the one idea of seeking extreme- or corner-point solutions.
- The premises of some classical algorithms that date back to the infancy days of OR are "challenged" with the presentation of algorithmic details that can enhance computational efficiency. For example, a generalized *artificial-variable-free* simplex algorithm is shown to apply any LP that starts both nonoptimal and infeasible.

- Algorithm "exchangeability" in solving OR problems from *distinct* optimization areas is presented where applicable throughout the book. The goal is to give the user more options for solving problems numerically. For example, two-person zero-sum games, and networks can be reformulated and solved as LPs.
- The *Aha! Moments* (first introduced in the tenth edition) are *math-free* stories, anecdotes, OR issues, and teaching methodology that delve into OR history and provide an appreciation of fundamental OR concepts.
- The TORA software is a cornerstone in understanding the algorithmic details. Its user manual is embedded directly in the software where and when needed during execution. TORA offers the unique *interactive* mode that allows the user to decide on the next calculation step, with instant feedback provided. If the input is correct, all (usually tedious) pertinent calculations are carried out, thus relieving the user of the burden of doing them manually; else, an appropriate error message is posted.
- Excel spreadsheets complementing TORA are used throughout the book; and, as in TORA, they engage the user interactively with immediate yes/no feedback regarding the user's choice of the next computational step. The spreadsheets automatically self-refresh when new input data is entered. The software can be found on the text's companion website at www.pearsonhighered.com/taha.
- The commercial software AMPL and Solver are used in example-format throughout the book. AMPL syntax is available in Appendix C.

Appended Table of Contents details the topics covered in the book. The following material highlights the specific changes/additions made in the eleventh edition.

1. **New Chapters:**
   - Chapter 1: Overview of OR, Analytics, AI, and ML in Decision-Making
   - Chapter 8: Stochastic Linear Programming
   - Chapter 14: Yield Management
2. **New Sections:**
   - Section 3.4.3: New Two-Phase Method with No Artificial Variables
   - Section 3.6.5: The 100% Rule of LP Sensitivity Analysis
   - Section 4.4.2: Generalized Simplex Algorithm
   - Section 4.5.4: Concurrent Changes in Feasibility and Optimality
   - Section 4.6: Transition from Textbook to Commercial Software Treatment of Sensitivity Analysis
   - Section 9.2.3: Benders' Decomposition Algorithm
   - Section 15.3: Bayes' Probabilities with ML Applications
3. **Completely Revised Chapter:**
   - Chapter 19: Discrete-Event and Monte Carlo Simulations
4. **Revised Sections:**
   - Section 3.6: Sensitivity Analysis
   - Section 4.5: Post-Optimal Analysis
   - Section 11.4.2: Reversal Heuristic

- Section 12.1: Recursive Nature of Dynamic Programming (DP) Computations
- Section 12.1.1 Recursive Equation and Principle of Optimality.
- Section 16.4: Ergodic (Regular) Markov Chain
- Section 21.1.1: Direct Search Method

5. **New Case Studies:**
   - One new case on analytics added in new Chapter 22
6. **New Aha! Moments:**
   - Four in Chapter 1, one in Chapter 6, one in Chapter 14, two in Chapter 15, two in Chapter 19, and one in Chapter 21
7. **Consolidated Chapters from Tenth Edition:**
   - ***Chapter 5: Transportation Model and Its Variants***
     10th—Chapter 5 + Appendix to Section 22.1 (Transshipment Model)
   - ***Chapter 6: Network Model***
     10th—Chapter 6 + Section 22.1 (Minimum-Cost Capacitated Flow Problem)
   - ***Chapter 7: Advanced Linear Programming***
     10th—Chapter 7 + Chapter 8 (Goal Programming) + Section 22.2 (Decomposition Algorithm) + Section 22.3 (Karmarkar Interior-Point Method)
   - ***Chapter 12: Dynamic Programming***
     10th—Chapter 12 (Deterministic Dynamic Programming) + Chapter 24 (Probabilistic Dynamic Programming)
   - ***New Chapter 13: Inventory Modeling***
     10th—Chapter 13 (Inventory Modeling) + Chapter 16 (Probabilistic Inventory Models)
   - ***New Chapter 16: Markov Chains***
     10th—Chapter 17 (Markov Chains)
   - ***New Chapter 17: Markovian Decision Process***
     10th—Chapter 25 (Markovian Decision Process)
   - ***New Chapter 22: Case Analysis***
     10th—Chapter 26 + one new case
8. **Renamed Appendixes:**
   - New appendix E: 10th—Chapter 14 (Review of Basic Probability)
   - New appendix F: 10th—Chapter 23 (Forecasting Models)
9. **General Editing:**
   The entire book has been edited to streamline the presentations and to remove outdated material. Another no-less-important goal of the new edition is to ensure gender neutrality and to eliminate the annoying editorial "we" throughout the book.
10. **Solutions Manual**
    Lastly, and by popular demand, I am happy to report that the Solutions Manual is now available in neatly typewritten format.
11. **Resources for Instructors**
    All instructor resources are available for download at www.pearson.com. If you are in need of a login and password for this site, please contact your local Pearson representative. The instructor resources include the following material: the Solutions Manual in PDF format and PowerPoint slides of all of the figures and tables in the text.

This page intentionally left blank

Acknowledgments

The 11th edition marks a 50-year milestone since the first publication of *Operations Research: An Introduction* by Macmillan in 1971. The book is currently published in 10 languages and I feel honored and humbled by the confidence, encouragement, and support I have received from my worldwide readership throughout the years.

I am grateful to the following colleagues for their valuable comments and suggestions: János D. Pintér (Lehigh University), Shokoufeh Mirzaei (California State Polytechnic University), Pomona Edward Huang (George Mason University), Yongjia Son (Clemson University), Adolfo R. Escobedo (Arizona State University).

I would be remiss if I did not thank Marcia Horton (Editor-in-Chief) and Holly Stark (Product Director) for their part in making the 50th anniversary a reality. Their confidence in the book (and in me), starting 35 years ago with the acquisition from Macmillan, was crucial in reaching the 50-year milestone. There are no words that can express my gratitude to Marcia and Holly for their support over the years.

Erin Ault (Manager of Content Strategy) spared no effort to provide support throughout the four years I spent preparing this 11th edition: securing timely peer reviews for the previous edition, ensuring thorough and smooth reviews of the manuscript files, skillfully managing a critical pagination issue during production without infringing on the content of the new edition, and, last but not least, taking the lead in selecting the most beautiful book cover.

I wish to thank Sandra Rodriguez (Managing Producer) and Devanshi Tyagi (Content Producer) for their contributions during the production process.

The dedicated effort, with great attention to detail, of Mahalakshmi Usha (Project Manager, Integra Software Services, India) was indispensable in converting my cluttered Word files into a polished 11th edition. *Thanks, Maha.*

I am grateful to Carrie Pennington, Ashley Reeves, and Mathew Sparks, of the University of Arkansas Industrial Engineering Department, for the immeasurable technical and editorial assistance they granted me during the four years I spent in the preparation of this edition.

I thank my dear friends Amaya and Froilán Cota who, with their language skills in both English and Spanish and a good ear for the nuance, have transformed my translation of the Dedication poem into vibrant English verses. *Mil gracias*.

H.A.T.
hat@uark.edu

This page intentionally left blank

About the Author

Hamdy A. Taha is Emeritus University Professor of Industrial Engineering with the University of Arkansas, where he taught and conducted research in operations research and simulation. He is the author of three other books on integer programming and simulation, and his works have been translated to numerous languages, including Chinese, Russian, and Spanish. He is also the author of book chapters, and his technical articles have appeared in *European Journal of Operations Research, IEEE Transactions on Reliability, IIE Transactions, Interfaces, Management Science, Naval Research Logistics Quarterly, Operations Research*, and *Simulation*.

Professor Taha was named a Senior Fulbright Scholar to Carlos III University, Madrid, Spain. He is fluent in three languages and has held teaching and consulting positions in Europe, Mexico, and the Middle East.

This page intentionally left blank

Trademarks

AMPL is a registered trademark of AMPL Optimization, LLC.
CPLEX is a registered trademark of ILOG, Inc.
Microsoft is a registered trademark and Windows and Excel are trademarks of Microsoft Corporation.
MINOS is a trademark of Stanford University.
Solver is a trademark of Frontline Systems, Inc.
TORA is a trademark of Hamdy A. Taha.

This page intentionally left blank

CHAPTER 1

Overview of OR, Analytics, AI, and ML in Decision-Making

Real-life Application—Crowdsourcing Analytics and OR Expertise . . . Two Studies with Different Outcomes.

Crowdsourcing, a word first coined in 2006, solicits the expertise of online community and outside sources worldwide to receive needed services, in place of adopting the traditional in-house group of experts. This real-life application, combining the use of analytics and OR tools, contrasts two experiences carried out by two global companies in different businesses: Netflix and Syngenta. One experience ended up in failure and the other in resounding success. Details of the study are given in Case 1, Chapter 22.

1.1 INTRODUCTION

This chapter introduces four prominent decision-making tools: OR, analytics, AI, and ML. These tools can be defined briefly as follows:

OR deals with *use of mathematical modeling to reach the best decision for analyzing operational systems subject to constrained resources*.

Analytics is rooted in *extracting new information from raw data using statistical analysis and other techniques for the purpose of reaching sound decision*.

AI develops *algorithms that use big data experiences to mimic human intelligence with the ultimate goal of automating complex tasks naturally performed by human beings*.

ML, an important branch of AI, *utilizes data analysis to automate the development of predictive models for the purpose of teaching machines how to make sound decisions with minimal or no human intervention*.

The definitions emphasize that **mathematical modeling** is the cornerstone of OR solutions. In analytics, AI, and ML, decision-making is **data** driven. Effectively, in OR the decision problem is translated into a (solvable) mathematical model representing a goal to be achieved subject to a set of restrictions; and in analytics, AI, and ML, analysis is data-driven, with the goal of discovering relationships that lead to informed decisions.

1.2 TWO DISTINCT APPROACHES FOR MAKING DECISIONS

OR seeks the *allocation* of constrained (limited) resources to competing activities for the purpose of **optimizing** a stated *objective*, such as *maximizing profit* or *minimizing cost*. It utilizes a *mathematical model* that expresses the objective and the constraints in terms of variables representing the unknowns of the decision situation. The resulting model is then solved numerically to determine the best values of the variables that optimize the objective function of the model. For example, in a product mix situation involving the use of limited raw materials, the variables are defined as the number of units to be manufactured of each product. A plausible objective in this situation is to maximize the revenue from producing the mix. The constraints ensure that the quantities produced are feasible within raw materials availability.

In recent years, massive amounts of data about all aspects of life are generated in multitudes of formats (e.g., text, numeric, image, video, and audio), buoyed by the internet, the smart phone, and other technological advances. The unprecedented advances in computations and the affordability of cloud computing make it possible to analyze available data into useful decision-making relationships, such as trends and correlations. Examples demonstrating the diversity of situations where analytics, AI, and ML applications include the following:

- Credit card companies *reduce fraud losses* by analyzing the purchasing habits of its card holders.
- Amsterdam, the bicycle capital of the world, *eases traffic and reduces pollution* by opportunely placing bike hubs in areas where they are most needed.
- Internet book-sellers *increase sales* by suggesting new titles based on patrons' reading habits and preferences.
- Trucking companies *reduce road accidents* by training drivers based on analyzing accident-causing data (e.g., driver fatigue, lane departure, and weather-related road conditions).
- Virtual assistants, such as Alexa, Bixby, Cortana, and Siri, use AI to answer questions at any time, *saving time and effort* in fetching needed information.
- Computer interface allows disabled humans to *regain lost physical functions.*
- Virtual tutor *expedites learning* by answering questions and providing meaningful feedback to students.
- Cars equipped with ML-based warning devices (e.g., lane departures and rear-end collisions) *make road driving safer and reduce fatalities and material losses*.
- ML-based apps *identify and quarantine harmful or time-wasting spam emails*.

In each of the preceding examples, the scope of the decision situation is so broad that it is impossible to represent them as traditional OR mathematical models. Yet, as conveyed by the italicized texts in these illustrations, analytics, AI, and ML embody a sense of **implicit optimization (**based on analyzing data) that is no less viable than the **explicit OR optimization** (based on mathematical modeling).

1.3 OR MATHEMATICAL MODELING

The first activities of OR started in England during World War II when teams of British scientists (ranging from pure and social sciences to engineering) set out to make learned decisions about the *best utilization* of war materiel. Following the end of the war, the ideas advanced in military operations were adapted to optimize operations in the civilian sector.

This section outlines how a mathematical model is developed and solved.

1.3.1 Elements of a Mathematical Model

Two simple decision situations are used to introduce the elements of OR mathematical modeling. The associated models happen to be *exact* representations of reality. The goal for the time being is to concentrate on the basics of mathematical modeling without the distraction of discussing model approximation.

Example 1.3-1 (Ticket Purchasing)

A businessperson has a 5-week commitment traveling between Fayetteville (FYV) and Denver (DEN). Weekly departure from Fayetteville occurs on Mondays for return on Wednesdays. A regular roundtrip ticket costs $400, but a 20% discount is granted when the roundtrip dates span a weekend. A one-way ticket in either direction costs 75% of the regular price. How should the tickets be bought for the 5-week period?

The model is constructed by answering three questions:

1. What are the decision **alternatives?**
2. Under what **restrictions** is the decision made?
3. What is an appropriate **objective criterion** for evaluating the alternatives?

The answer to the first question can include three plausible alternatives:

Alternative 1: Buy five regular FYV-DEN-FYV for departure on Monday and return on Wednesday of the same week.

Alternative 2: Buy one FYV-DEN for week 1, four DEN-FYV-DEN spanning weekends, and one return DEN-FYV for week 5.

Alternative 3: Modify Alternative 2 so that *each* roundtrip ticket would span a weekend by buying one FYV-DEN-FYV for Monday of week 1 and Wednesday of week 5 and four DEN-FYV-DEN to cover the remaining travel legs.

The travel restriction in all three alternatives is to leave FYV on each Monday for return on Wednesday of the same week over a span of 5 weeks.

A logical criterion for evaluating the proposed alternatives is the price paid for all tickets. The alternative yielding the smallest cost is the **optimum** (best).[1] Specifically,

$$\text{Alternative 1 cost} = 5 \times 400 = \$2000$$
$$\text{Alternative 2 cost} = .75 \times 400 + 4 \times (.8 \times 400) + .75 \times 400 = \$1880$$
$$\text{Alternative 3 cost} = 5 \times (.8 \times 400) = \$\mathbf{1600}$$

Alternative 3 is the cheapest and hence is the **optimum solution**.

Remarks. Though OR models are designed to *optimize* a specific objective criterion subject to a set of constraints, the *quality* of the optimum solution depends on the accuracy of the model in representing the real system. In the present ticket purchasing model, if *all* the relevant alternatives for purchasing the tickets were not identified, then the resulting solution would be optimum only relative to the alternatives represented in the model. To be specific, identifying alternative 3 may be a bit "tricky" and hence can be overlooked. Should that happen, the resulting solution would be **suboptimal** because it would call for purchasing the tickets for $1880, per alternative 2.

Though the tickets example illustrates the three main components of the OR model—alternatives, objective criterion, and constraints—situations differ in the details of how each component is developed and how the resulting model is solved, as the following example demonstrates.

Example 1.3-2 (Garden Fence)

Imagine that you have a specified amount of construction material to fence a rectangular backyard vegetable garden of perimeter L ft. How should the fence be constructed to *maximize* the enclosed area (presumably to *maximize* the garden yield)?

In contrast with the tickets example, where the number of alternatives is finite (= 3), the present example theoretically boasts an infinite number of alternatives; meaning, the *width* and *height* of the rectangle can each assume infinity of values between 0 and $\frac{L}{2}$. In this regard, the *width* and the *height* are **continuous variables**.

The fence problem can be translated into a **mathematical model** in the following manner: For a specified perimeter, the *width* and *height* are the **unknown variables** whose values are determined by solving the following exact model:

$$\textit{Maximize}\ \text{area} = \text{width} \times \text{height}$$

subject to

$$(\text{width} + \text{height}) = \frac{\text{perimeter}}{2}$$

$$\text{width} \geq 0,\ \text{height} \geq 0$$

The constraints *width* ≥ 0 and *height* ≥ 0 are **nonnegativity conditions** and are assumed to hold in this situation because the dimensions of the rectangle cannot be negative. There are situations, however, where the variables may assume positive, zero, or negative values (e.g., temperature), in which case the variables become **unrestricted in sign.**

[1]Intuitively, and without evaluating the alternatives, alternative 1 is the most expensive and alternative 3 is the cheapest.

Though all mathematical programming languages (e.g., AMPL) allow self-documentation by using explicit naming (such as *area*, *width*, *height*, and *perimeter*), it is more convenient in textbook setting to use compact algebraic symbols.

Define

$$w = \text{width of the rectangle in ft}$$
$$h = \text{height of the rectangle in ft}$$
$$z = \text{area in ft}^2$$

Given the perimeter of the fence is 100 ft, the complete mathematical model can then be presented as

$$\text{Maximize } z = wh$$

subject to

$$w + h = 50$$
$$w, h \geq 0$$

Calculus can be used to find the optimum solution of this model. Given $w = 50 - h$, then $z = wh = (50 - h)h = 50h - h^2$. Using calculus, the necessary condition yields

$$\frac{dz}{dh} = 50 - 2h = 0 \Rightarrow h = 25$$

This condition is sufficient because z is concave (see Chapter 20), yielding the optimum solution $h = 25\text{ft}$, with $w = 50 - h = 25\text{ft}$ with $z = 625\text{ ft}^2$. The answer asserts that a square of side $\frac{L}{2}$ provides the maximum garden area.[2]

The format of the mathematical model developed for the Fence Problem is typical of the way OR models are presented pictorially. Of course, the objective function can be either **maximized** or **minimized** depending on the nature of the decision situation (e.g., profit is maximized, and cost is minimized). Figure 1.1 provides the general layout of the OR mathematical model.

FIGURE 1.1

Layout of OR mathematical model

OR Mathematical Model
Optimize (maximize or minimize) **Objective** subject to **Constraints**

[2]This situation paraphrases a classic mathematical problem solved by the famed Greek "father of geometry" Euclid (born c. 300 BC in Alexandria, Egypt). In those days, algebra (developed c. 800 AD) was unknown, and Euclid relied on geometric axioms to prove that *among all rectangles of a fixed perimeter, a square shape yields the largest area*. Problem 1-3 is a brainteaser based on Euclid's geometric proof.

A solution is **feasible** if it satisfies all the constraints. It is **optimal** if, in addition to being feasible, it yields the best (maximum or minimum) value of the objective function.

Aha! Moment. Nature's Ultimate Optimizers, or How Bees Construct Honeycombs![3]

Every time I come across an image of a honeycomb, I am in awe of the perfect hexagonal design of the compartments "architected" and constructed by nearly 50,000 worker bees per hive to store honey. But why hexagons? This question has intrigued mathematicians for over 2000 years. In 36 BC, the Roman scholar Marcus Terentius Varro *hypothesized* that a hexagonal design is the optimum way to divide a surface into equal areas with minimum total perimeter. Varro's proposal, dubbed the **honeycomb conjecture**, remained just that – a conjecture – until 1999, when it was proven by American mathematician Thomas C. Hales while at the University of Michigan.

Why are hexagons optimal? The answer lies in economizing three variables: space usage, amount of wax used, and volume of the cell. Space economy dictates compactness of the cells with no unused gaps in the structure. Among all equal-sided two-dimensional geometric shapes, only triangles, squares, and hexagons meet this criterion. Of these three shapes, the hexagon is the best because, as mathematically proven by Hales, it produces the smallest total perimeter, hence uses the least amount of wax (the triangle is the worst).[4] Moreover, a hexagon-based cell has the largest storage volume among the three shapes. An additional bonus is that in architecture the hexagon is the strongest shape known under pressure.

Remarkably, bees further optimize the honeycomb structure by scaling the size of each hexagonal cell to match the size of the colony bees!

1.3.2 Solving the Mathematical Model

In practice, OR does *not* offer a "one-size-fits-all" method for solving all mathematical models. Instead, the type and complexity of the mathematical model dictate the nature of the solution method. An oddity of most OR techniques is that solutions are not generally obtained in (formula-like) closed forms. Instead, they are determined by an **algorithm**, defined by Merriam-Webster Dictionary as *A step-by-step procedure for solving a problem or accomplishing some end.* It provides a set of computational rules that are applied repetitively to the problem, with each repetition (called **iteration**) moving the answer closer to the optimum value. Because the computations in each iteration are typically tedious and voluminous, in most algorithms the use of computers is imperative.

The most prominent (and oldest) OR algorithm is **linear programming**, designed for models with linear objective and constraint functions. Other algorithms

[3]Robert Krulwich, What Is It About Bees and Hexagons?, *NPR*, aired on May 14, 2013, 9:50 AM ET, accessed July 16, 2017.

[4]Hales, Thomas C., "The Honeycomb Conjecture". *Discrete and Computational Geometry*, Vol. 25, No. 1, 2001, pp. 1–22.

include **integer programming** where variables must assume integer values, **dynamic programming** in which the original model is decomposed into smaller (more computationally-manageable) subproblems, **network programming** in which the problem are modeled as a network, and **nonlinear programming** in which functions of the model are nonlinear. These algorithms are but a few of the arsenal of OR solution methods.

Some mathematical models may be so complex that it is impossible to solve them by available optimization algorithms. In such cases, it may be necessary to settle for a *good* solution using **heuristics** or **metaheuristics**: a collection of intelligent search *rules of thumb* that move the solution point advantageously toward the optimum, but not necessarily achieving optimality.

Aha! Moment: First Ever Algorithm Programmer—Ada Lovelace.[5]

Though the first conceptual development of an algorithm is attributed to the founder of algebra Muhammad Ibn-Musa Al-Khwarizmi (born c. 780 in Khwarazm, Uzbekistan, died c. 850 in Baghdad, Iraq)[6], it was British Ada Lovelace (1815–1852) who developed the first *computer* algorithm. And when speaking of computers, it is the mechanical Difference and Analytical Engines pioneered and designed by British mathematician Charles Babbage (1791–1871).[7]

Lovelace had keen interest in mathematics. As a teenager, she visited Babbage home and was fascinated by his invention and its potential uses in doing more than just arithmetic operations. Collaborating with Babbage, she translated into English an article that provided the design details of the Analytical Engine. The article was based on lectures Babbage presented in Italy. In the translated article, Lovelace appended her own notes (which turned out to be longer than the original article and included some corrections of Babbage's design ideas). One of her notes detailed the first-ever *algorithm*, that of computing Bernoulli numbers on the yet-to-be-completed Analytical Engine. She even predicted that Babbage machine would have the potential to manipulate symbols (not just numbers) and to create complex music scores.

In Lovelace's honor, the computer language *Ada* (developed for the United States Department of Defense) was named after her. The annual mid-October *Ada Lovelace Day* is an international celebration of women in science, technology, engineering, and mathematics (STEM). And those of us who visited St. James Square in London may recall the blue plaque that reads "Ada Countess of Lovelace (1815 — 1852) Pioneer of Computing."

[5]https://en.wikipedia.org/wiki/Ada_Lovelace, accessed October 13, 2015.

[6]According to Dictionary.com, the word *algorithm* originates "from Medieval Latin algorismus, a mangled transliteration of Arabic Al-Khwarizmi."

[7]Lack of funding, among other factors, prevented Babbage during his lifetime from building fully working machines. It was only in 1991 that the London Science Museum built a complete Difference Engine No. 2 using the same materials and technology available to Babbage, thus vindicating his design ideas. There is currently an ongoing long-term effort to construct a fully working Analytical Engine funded entirely by public contributions. It is impressive that modern-day computers are based on the same principal components (memory, CPU, input, and output) advanced by Babbage 100 years earlier.

1.3.3 Proxy Optimization – Queuing, Simulation, and Monte Carlo Models

Queuing theory, one of the oldest techniques in OR, provides **probability-based models** (where arrivals and/or service times at a service facility are typically random) that deal with the phenomenon of waiting, an every-day experience where servers attend to customers (see Chapter 18). In the familiar sense, customers and servers are real human beings. They also can be automated versions of a customer (e.g., telephone calls arriving at a telephone exchange) or a server (e.g., a bank ATM machine).

Faster service (requiring hiring more servers) is desirable from customer's standpoint (less wait) but could be economically costly for the facility operator, and vice versa. Hence is the need for a *compromise* design of a service facility that *both* the customer and the server can live with. To that end, queuing, models provide statistical measures, such as the average waiting time and the average utilization of a server, that assess "compromise" solutions from the (conflicting) standpoints of both the customer and the server.

Simulation modeling deals with the waiting phenomenon as well. It differs from queuing in that it uses the computer to *mimic* the actions associated with the time events of arrival at and departure from a service facility. As new events occur, pertinent data are collected for the purpose of computing desired measures of performance for the system (see Chapter 19). In a way, simulation may be regarded as the next best thing to observing a real system. Unlike queuing models where mathematical restrictions could limit their applicability in practice, simulation modeling is flexible because it can represent (just about) any waiting line situation. The disadvantage is that a simulation model takes time and effort to develop and their execution on the computer could be lengthy. Additionally, simulation is basically a *statistical experiment* whose observations must be interpreted statistically, expressing the output in descriptive statistics, such as averages, standard deviations, and confidence intervals, among others.

Monte Carlo method, a forerunner to present-day simulation, is a class of computational algorithms that uses random sampling to obtain numerical estimates of mostly deterministic parameters such as estimating the constant π ($\cong$ 3.14159), matrix inversion, or the value of a hard to evaluate integrals of complex mathematical functions. The final estimate is usually presented as confidence intervals and histograms.

Queuing, simulation, and Monte Carlo optimization is not *explicit* in the sense used in OR. Rather, it is in line with the *implicit* sense used in analytics, AI, and ML, alluded to earlier in this chapter. Their measures of performance play a *proxy* role in "what-if" experimentation for the purpose of designing efficient service facilities (see Problem 1-13).

1.3.4 OR Is More Than Just Mathematics

Mathematical modeling is a cornerstone of OR. Yet, "commonsense" non-mathematical approaches can sometimes lead to simpler but effective solutions. Six illustrations are presented here in support of this argument.

1. The stakes were high when United Parcel Service (UPS) unrolled in 2004 its ORION software (based on the sophisticated Traveling Salesperson

Algorithm—see Chapter 11) to provide its drivers with tailored daily delivery itineraries. The software generally proposed shorter routes than those presently taken by the drivers, with potential savings of millions of dollars a year in fuel cost. For their part, the drivers resented the notion that a machine can "best" them, given their long years of experience on the job. Faced with this human dilemma, ORION developers resolved the issue simply placing a visible banner on the itinerary sheets that read "Beat the Computer." At the same time, they kept ORION-generated routes intact. The drivers took the challenge to heart, with some of them beating computer suggested route. ORION was no longer "putting them down." Instead, they regarded the software as complementing their intuition and experience.[8]

2. Travelers arriving at the Intercontinental Airport in Houston, Texas, complained about the long wait for their baggage after landing. Authorities increased the number of baggage handlers in hope of alleviating the problem, but complaints persisted. In the end, a decision was made to simply move arrival gates farther away from baggage claim area, forcing the passengers to walk longer distances before reaching the baggage area. The complaints disappeared because the extra walking allowed ample time for the luggage to be delivered to the carousel area.[9]
3. In a study of the check-in counters at a large British airport, a U.S.–Canadian consulting team used queuing theory to investigate and analyze the situation. Part of the solution recommended the use of well-placed signs urging passengers within 20 minutes of departure time to advance to the head of the queue and request priority service. The solution was not successful because the passengers, being mostly British, were "conditioned to strict queuing behavior." Hence, they were reluctant to move ahead of others waiting in the queue.[10]
4. In a steel mill in India, ingots were first produced from iron ore and then used in the manufacture of steel bars and beams. The manager noticed a long delay between the ingots production and their transfer to the next manufacturing phase (where end products were produced). Ideally, to reduce reheating cost, manufacturing should start soon after the ingots had left the furnaces. Initially the problem was perceived as a line-balancing situation, which could be resolved either by reducing the output of ingots or by increasing the manufacturing capacity. Instead, the OR team used simple charts to summarize the output of the furnaces during the three shifts of the day. They discovered that during the third shift starting at 11:00 P.M., most of the ingots were produced between 2:00 and 7:00 A.M. Investigation revealed that third-shift operators, having hours to spare to meet their quota, preferred to get long periods of rest at the start of the shift and then make up for lost production during morning hours. The problem was solved by

[8]http://www.fastcompany.com/3004319/brown-down-ups-drivers-vs-ups-algorithm. See also "At UPS, the Algorithm Is the Driver," *Wall Street Journal*, February 16, 2015.

[9]A. Stone, "Why Waiting Is Torture," *New York Times*, August 18, 2012.

[10]A. Lee, *Applied Queuing Theory*, St. Martin's Press, New York, 1966.

"leveling out" both the number of operators and the production schedule of ingots throughout the shift.

5. Responding to complaints of slow elevator service in a large office building, the OR team initially perceived the situation as a waiting-line problem that might require the use of mathematical queuing analysis or simulation. After studying the behavior of the people waiting for the service, the psychologist on the team suggested installing full-length mirrors at the entrance to the elevators. The complaints disappeared as people were kept occupied watching themselves and others while waiting for the elevator.
6. Departments in a production facility share the use of three trucks to transport material. Requests initiated by a department are filled on a first-come-first-serve basis. Some departments complained of long wait for service, and demanded adding a fourth truck to the pool. Ensuing simple tallying of the usage of the trucks showed modest daily utilization, eliminating the need for a fourth truck. Further investigations revealed that the trucks were parked in an obscure parking lot out of the line of vision for the departments. A requesting supervisor, lacking a sighting of the trucks, assumed that no trucks were available and hence did not initiate a request. The problem was solved by installing two-way radio communication between the truck lot and each department.[11]

Four conclusions can be drawn from these illustrations:

1. The OR team should explore the possibility of using "different" ideas to resolve situations. The (common-sense) solutions proposed for the UPS problem (using *Beat the Computer* banner to engage drivers), the Houston airport (moving arrival gates away from the baggage claim area), and the elevator problem (installing mirrors) are rooted in human psychology rather than in mathematical modeling. This is the reason OR teams may generally seek the expertise of individuals trained in social science and psychology, a point that was recognized and implemented by the first OR team in Britain during World War II.
2. Before jumping to the use of sophisticated mathematical modeling, a bird's eye view of the situation should be adopted to uncover possible nontechnical reasons that led to the problem in the first place. In the steel mill situation, this was achieved by using only simple charting of the ingots production to discover the imbalance in the third-shift operation; and that was all that was needed to resolve the issue. A similar simple observation in the case with the transport tucks situation also led to a simple solution of the problem.
3. An OR study should not start with a bias toward using a specific mathematical tool before the use of the tool is justified. For example, because linear programming (Chapter 2 and beyond) is a successful technique, there may be a

[11] G. P. Cosmetatos, "The Value of Queuing Theory—A Case Study," *Interfaces*, Vol. 9, No. 3, pp. 47–51, 1979.

tendency to use it as the modeling tool of choice. Such an approach may lead to a mathematical model far removed from the real situation. It is thus imperative to analyze available data, initially using the simplest possible techniques, to understand the essence of the problem. Once the problem is defined, a decision can be made regarding the most appropriate solution tool. In the steel mill problem, simple charting of the ingots production was all that was needed to clarify the situation.

4. Solutions are rooted in people and not in technology. Any solution that does not take human behavior into consideration is apt to fail. Even though the solution of the British airport problem may have been mathematically sound, the fact that the consulting team was unaware of the cultural differences between the United States and Britain (Americans and Canadians tend to be less formal) resulted in an un-implementable recommendation. The same viewpoint can, in a way, be expressed in the UPS case.

1.4 ANALYTICS MODELING

Unlike OR, which is driven by mathematical optimization models, analytics is rooted in analyzing data. In fact, some proponents of analytics predict that "the most important raw material of the twenty-first century is data. Data are the new oil."[12] And just as crude oil must be *refined* into useful end products, raw data must be *transformed* to uncover patterns and trends for making better decisions.

Aha! Moment. The Value of Data… or How a Car Dealership Embraced "Analytics" Long before the PC Era!

Years ago, in the 70's, I visited a car dealership to buy a car and ended up chatting with the owner. What impressed me most was the way the owner decided which radio stations to select for advertising. Over the years, in pre-PC era, the dealership collected and updated data about the radio stations customers listened to, not by using questionnaires, but by instructing the shop mechanics to scan the radios of cars brought to service for the area stations customers had preselected as favorites. Collected data was tallied to decide which radio stations should be targeted for advertising. When asked about the secret of the dealership success, the owner's standard response was "I advertise smart," without divulging details.

I remember thinking that the owner was a shrewd businessperson for recognizing the power of data in making better decisions, in effect employing a variation of present-day analytics at a time when such a tool (and computers) were practically nonexistent.

1.4.1 Elements of the Analytics Model

The following example is a simplified but instructive application of the analytics model.

[12] A. Weigend, *Data for the People*, Basic Books, Ney York, 2017, p. 13.

Example 1.4-1 (Credit Card Fraud Prevention)

One of the earliest successful applications of analytics deals with credit card fraud. I recall years ago receiving a midnight telephone call from my credit card company inquiring if my daughter was authorized to withdraw cash against my account. I knew that the transaction was fraudulent because my daughter was only 3 years old at the time. I was curious, though, to learn how the company was alerted to the possibility of a fraud. I was told that their system continually looks for patterns/trends and search for anomalies based on my card use history (i.e., raw data) to classify and update my purchasing habits, and that withdrawing cash from my account was not one of them.

The example above reveals the essence of analytics: Raw data, represented by past credit card transactions is processed in two phases: In Phase 1, the data is explored to uncover patterns and trends that **describe** past purchasing behavior. Phase 2 then uses the *transformed* data to **predict** if a future purchase deviates from past purchasing habits, raising a red flag and, if warranted, taking appropriate corrective actions.

In some situations, the complete analytics model can include a third phase, based on the results from phases 1 and 2, to **prescribe** a good, if not OR-optimum, decision for the problem.

Figure 1.2 summarizes the three phases of the analytics model.

1.4.2 Data Relevance in Analytics

Although analytics thrives on analyzing lots of data, the golden rule is that *volume is no substitute for quality*; meaning, collected data must be *relevant* to the decision situation. Thus, it is necessary to identify the type and amount of raw data to be collected.

How is data relevance ascertained? In some situations, the answer may be immediately obvious. For example, in the credit card fraud situation (Example 1.1-1), card transactions (e.g., purchases) provide the main raw data needed to develop the analytic model. In other more complex situations (e.g., reduction of the number of large-trucks road accidents), the type of data needed for developing the analytics model is not as obvious. One approach calls for *reverse-scanning* of the phases of the analytics model; that is,

$$\text{Goal} \rightarrow \text{Causes} \rightarrow \text{Effects} \rightarrow \text{Data}$$

The idea is that the goal of the decision situation is impacted by the *causes* that necessitate making the decision in the first place. Next, the causes are manifested in terms of *effects*, which then reveal the type of data needed for the decision situation.

FIGURE 1.2

Phases of the Analytics Model

Phases of Analytics Model
Phase 1: Explore the raw data to **describe** the past and present.
Phase 2: Extrapolate the past and present to **predict** what *might* happen in the future and take corrective actions, if necessary.
Phase 3: Use OR optimization models, based on the results from phases 1 and 2, to **prescribe** an informed decision.

A simplified illustration deals with increasing revenues in concession stands in a theme park:

Goal: Improved revenues

Causes: Location, menu, hours of operation, and weather.

Effects → data: (Proximity to park entrance/exit or to specific park themes) → (types of food/drinks/ snacks served, business hours for the stand, quantities sold, seasonal weather conditions).

1.4.3 Example Applications

This section presents two representative applications: The first demonstrates a decision situation that uses phases 1 and 2 only of the analytics model, and the second illustrates how, per phase 3, an additional OR algorithm is used to reach an optimum decision.

Large Trucks Road accidents. Drivers of large trucks are subject to fatigue, distraction, adverse road conditions, and other unexpected events that could result in road accidents. Statistics in the United States indicate that the cost of accidents involving large trucks exceeded $100 billion in 2014. How can the cost of accidents involving large trucks be reduced?

The OR mathematical model (e.g., the garden fence problem of Example 1.3-1) and the present situation share the common economic goal of *improving* their respective outcomes: The former optimizes an objective function and the latter reduces the cost of accidents. But this is about the extent of the "similarities" between the two modeling situations; in the sense that the OR concepts of using an *objective* and dealing with *limited resources* are not well-defined in the truck accident case. Instead, relevant data about how, why, where, and when accidents occur are analyzed via appropriate analytics with the goal of taking corrective actions to reduce road accidents.

The development of the analytics model for large trucks road accidents involves three aspects:

1. Relevance of data.
2. Descriptive analytics.
3. Predictive analytics.

Data Relevance. Application of the reverse scanning scheme (Section 1.4.2) determines a list of the key causes and effects given in Table 1.1.

The *effects* column in Table 1.1 points to the data needed to analyze the current situation. Safety infractions provide a type of data needed to analyze and make predictions about driver behavior, and records of engine maintenance and travel routes reveal information about engine/road maintenance. Weather forecast provide the data needed to predict the "drivability" condition of the roads.

TABLE 1.1 List of causes and effects in large truck accidents

Causes[a]	Effects
Driver behavior	Fatigue, emotional stress, distraction, sudden acceleration, speeding, hard braking, swerves, lane departures, excessive fuel consumption.
Road conditions	Maintenance, upgrades, road curves, slopes, and rest areas.
Engine condition	Periodic/preventive maintenance.
Weather conditions	Inclement weather, poor visibility.

[a] According to published statistics, driver behavior is responsible for approximately 80% of road accidents.

Descriptive Phase. Some of the data (such as hard breaking, swerving, and fuel consumption) is gathered and transmitted in real time using onboard telematics, enriched by video feeds from cameras facing both the road and truck interior. Other data (such as weather, engine maintenance, and road conditions) is collected from other sources. Whatever the source, the data is tested for reliability and validity to weed out incorrect or incomplete sets before being incorporated in the decision process.

Prediction Phase. An important goal of analyzing data is to recommend ways to reduce accidents by concentrating on drivers in need of attention based on history of safety infractions, including real-time monitoring, and/or providing coaching opportunities. For this purpose, raw data is analyzed using appropriate statistical tools that can be as simple as comparing alternatives using percentages, averages, and histograms. In some situations, analytics modeling employs advanced statistical analysis aimed at revealing hidden correlations. For example, one such study reveals that "The segment of drivers seen as most likely to be in a collision also consume over 7.5 percent more fuel."[13]

Regardless of the level (simple or advanced) at which data is analyzed, the use of an appropriate software handling large sets of data is indispensable. The most widely available system is the Excel spreadsheet, with statistical tools ranging from basic descriptive statistics to regression, hypothesis testing, sampling, and experimental design. Another popular (and free) software is the open source R programming language.

Inventory Balancing—Analytics with Embedded Prescriptive OR Model. Retailers aspire to stock the right item at the right place for the right time. When a retailer operates multiple stores, seasonal items sell at different rates in different locations. Toward the end of the season, some locations may end up with surplus inventory and others may run out of stock. Both situations are undesirable because they entail lost revenue: Surplus inventory is usually heavily discounted for quick sales and shortages mean lost

[13] Using Big Data And Predictive Analytics To Predict Which Truck Drivers Will Have An Accident. https://www.forbes.com/sites/stevebanker/2016/10/18/using-big-data-and-predictive-analytics-to-predict-which-truck-drivers-will-have-an-accident/?sh=40651d401cb0

sales and potential customer dissatisfaction. A plausible remedy for this situation is to even out (or balance) end-of season inventory positions in the different locations by moving items from low-demand outlets to high-demand ones. The proposed movement of merchandise among stores involve the additional cost of shipping and handling, giving rise to a companion optimization situation that calls for minimizing the associated transportation costs.

Inventory control is one of the oldest OR models (see Chapter 13). The basis of the mathematical model is that both surplus and shortage do occur in business operation, hence the objective in most cases is to determine the optimum inventory level that minimizes the sum of the conflicting costs of surplus and shortage inventories.

In the current inventory-balancing situation, the problem is more involved, requiring a two-step analysis:

Step 1: Prediction of which stores will have surplus and which will experience shortage as the season nears an end.

Step 2: Determination of the most economical plan to transport units form surplus to shortage locations.

The first step is rooted in forecasting analytics (phases 1 and 2 in Figure 1.2). It uses historical demand data to develop a forecasting model (see Appendix F) to predict demand. This prediction is the basis for projecting end-of-season surplus or shortage amounts at each location.

Step 2 is a classical OR model known as the transportation model (Chapter 5). It deals with attempting to satisfy requests at destinations subject to availabilities at sources, shipping units from *sources* (surplus stores) to *destinations* (shortage stores). Associated with each transportation route from a source to a destination is a unit transportation cost. The goal is to transport the inventory item from sources to destinations at the least possible cost.

Example 1.4-2

The following data for a hypothetical 5-store situation is used to demonstrate the development of the OR model in Step 2:

Store	S1	S2	S3	S4	S5
Starting inventory	100	100	100	100	100
Projected end of season demand	50	120	70	140	65
Projected surplus	50		30		35
Projected shortage		20		40	

Data in the second row are computed from analytics forecasting models. The surplus and shortage rows are then computed from data in the first two rows.

Figure 1.3 summarizes the resulting transportation model. Stores 1, 3, and 5 are shipping sources, whereas stores 2 and 4 are the receiving destinations. The arrows between