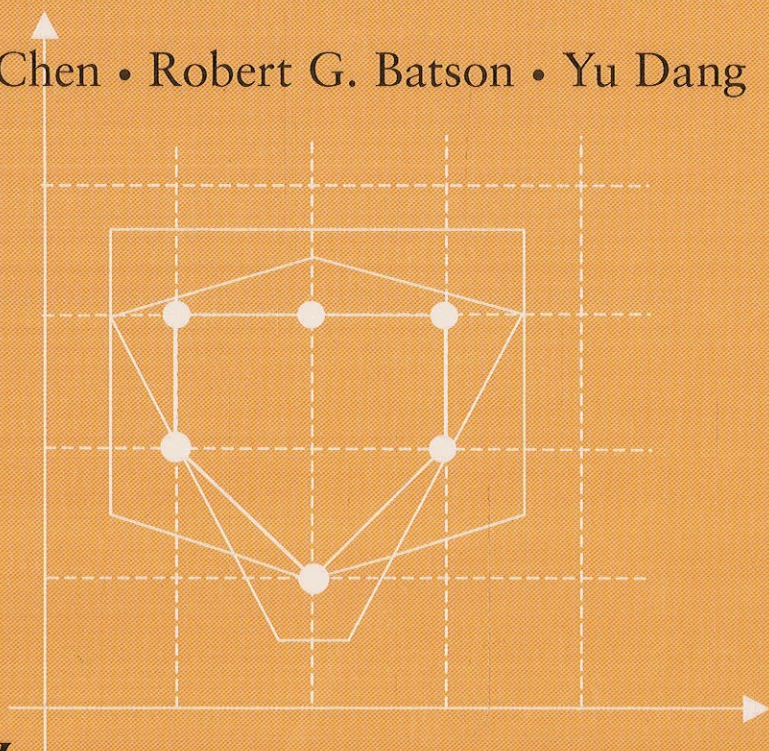


Applied Integer Programming

MODELING AND SOLUTION

Der-San Chen • Robert G. Batson • Yu Dang



CONTENTS

PREFACE

xvii

PART I MODELING

1

1 Introduction

3

- 1.1 Integer Programming, 3
- 1.2 Standard Versus Nonstandard Forms, 5
- 1.3 Combinatorial Optimization Problems, 7
- 1.4 Successful Integer Programming Applications, 8
- 1.5 Text Organization and Chapter Preview, 8
- 1.6 Notes, 17
- 1.7 Exercises, 18

2 Modeling and Models

21

- 2.1 Assumptions on Mixed Integer Programs, 22
- 2.2 Modeling Process, 28
- 2.3 Project Selection Problems, 30
 - 2.3.1 Knapsack Problem, 30
 - 2.3.2 Capital Budgeting Problem, 31
- 2.4 Production Planning Problems, 32
 - 2.4.1 Uncapacitated Lot Sizing, 33
 - 2.4.2 Capacitated Lot Sizing, 34
 - 2.4.3 Just-in-Time Production Planning, 34

- 2.5 Workforce/Staff Scheduling Problems, 36
 - 2.5.1 Scheduling Full-Time Workers, 36
 - 2.5.2 Scheduling Full-Time and Part-Time Workers, 37
- 2.6 Fixed-Charge Transportation and Distribution Problems, 38
 - 2.6.1 Fixed-Charge Transportation, 38
 - 2.6.2 Uncapacitated Facility Location, 40
 - 2.6.3 Capacitated Facility Location, 41
- 2.7 Multicommodity Network Flow Problem, 41
- 2.8 Network Optimization Problems with Side Constraints, 43
- 2.9 Supply Chain Planning Problems, 44
- 2.10 Notes, 47
- 2.11 Exercises, 48

3 Transformation Using 0–1 Variables

54

- 3.1 Transform Logical (Boolean) Expressions, 55
 - 3.1.1 Truth Table of Boolean Operations, 55
 - 3.1.2 Basic Logical (Boolean) Operations on Variables, 56
 - 3.1.3 Multiple Boolean Operations on Variables, 58
- 3.2 Transform Nonbinary to 0–1 Variable, 58
 - 3.2.1 Transform Integer Variable, 58
 - 3.2.2 Transform Discrete Variable, 60
- 3.3 Transform Piecewise Linear Functions, 60
 - 3.3.1 Arbitrary Piecewise Linear Functions, 60
 - 3.3.2 Concave Piecewise Linear Cost Functions:
Economy of Scale, 63
- 3.4 Transform 0–1 Polynomial Functions, 64
- 3.5 Transform Functions with Products of Binary and Continuous Variables: Bundle Pricing Problem, 66
- 3.6 Transform Nonsimultaneous Constraints, 69
 - 3.6.1 Either/Or Constraints, 69
 - 3.6.2 p Out of m Constraints Must Hold, 70
 - 3.6.3 Disjunctive Constraint Sets, 71
 - 3.6.4 Negation of a Constraint, 71
 - 3.6.5 If/Then Constraints, 71
- 3.7 Notes, 72
- 3.8 Exercises, 73

4 Better Formulation by Preprocessing

79

- 4.1 Better Formulation, 79
- 4.2 Automatic Problem Preprocessing, 86
- 4.3 Tightening Bounds on Variables, 87
 - 4.3.1 Bounds on Continuous Variables, 87
 - 4.3.2 Bounds on General Integer Variables, 88
 - 4.3.3 Bounds on 0–1 Variables, 90

- 4.3.4 Variable Fixing, Redundant Constraints, and Infeasibility, 91
- 4.4 Preprocessing Pure 0–1 Integer Programs, 93
 - 4.4.1 Fixing 0–1 Variables, 93
 - 4.4.2 Detecting Redundant Constraints And Infeasibility, 95
 - 4.4.3 Tightening Constraints (or Coefficients Reduction), 96
 - 4.4.4 Generating Cutting Planes from Minimum Cover, 97
 - 4.4.5 Rounding by Division with GCD, 98
- 4.5 Decomposing a Problem into Independent Subproblems, 99
- 4.6 Scaling the Coefficient Matrix, 100
- 4.7 Notes, 101
- 4.8 Exercises, 101

5 Modeling Combinatorial Optimization Problems I

105

- 5.1 Introduction, 105
- 5.2 Set Covering and Set Partitioning, 106
 - 5.2.1 Set Covering Problem, 107
 - 5.2.2 Set Partitioning and Set Packing, 111
 - 5.2.3 Set Covering in Networks, 111
 - 5.2.4 Applications of Set Covering Problem, 113
- 5.3 Matching Problem, 115
 - 5.3.1 Matching Problems in Network, 115
 - 5.3.2 Integer Programming Formulation, 116
- 5.4 Cutting Stock Problem, 117
 - 5.4.1 One-Dimensional Case, 117
 - 5.4.2 Two-Dimensional Case, 120
- 5.5 Comparisons for Above Problems, 121
- 5.6 Computational Complexity of COP, 121
 - 5.6.1 Problem Versus Problem Instance, 123
 - 5.6.2 Computational Complexity of an Algorithm, 123
 - 5.6.3 Polynomial Versus Nonpolynomial Function, 124
- 5.7 Notes, 125
- 5.8 Exercises, 126

6 Modeling Combinatorial Optimization Problems II

130

- 6.1 Importance of Traveling Salesman Problem, 130
- 6.2 Transformations to Traveling Salesman Problem, 133
 - 6.2.1 Shortest Hamiltonian Paths, 133
 - 6.2.2 TSP with Repeated City Visits, 134
 - 6.2.3 Multiple Traveling Salesmen Problem, 135
 - 6.2.4 Clustered TSP, 137
 - 6.2.5 Generalized TSP, 137
 - 6.2.6 Maximum TSP, 139

- 6.3 Applications of TSP, 139
 - 6.3.1 Machine Sequencing Problems in Various Manufacturing Systems, 140
 - 6.3.2 Sequencing Problems in Electronic Industry, 140
 - 6.3.3 Vehicle Routing for Delivery/Dispatching, 141
 - 6.3.4 Genome Sequencing for Genetic Study, 142
- 6.4 Formulating Asymmetric TSP, 142
 - 6.4.1 Subtour Elimination by Dantzig–Fulkerson–Johnson Constraints, 143
 - 6.4.2 Subtour Elimination by Miller–Tucker–Zemlin (MTZ) Constraints, 144
- 6.5 Formulating Symmetric TSP, 146
- 6.6 Notes, 148
- 6.7 Exercises, 149

PART II REVIEW OF LINEAR PROGRAMMING AND NETWORK FLOWS

153

7 Linear Programming—Fundamentals

155

- 7.1 Review of Basic Linear Algebra, 155
 - 7.1.1 Euclidean Space, 155
 - 7.1.2 Linear and Convex Combinations, 156
 - 7.1.3 Linear Independence, 156
 - 7.1.4 Rank of a Matrix, 156
 - 7.1.5 Basis, 157
 - 7.1.6 Matrix Inversion, 157
 - 7.1.7 Determinant of a Matrix, 157
 - 7.1.8 Upper and Lower Triangular Matrices, 158
- 7.2 Uses of Elementary Row Operations, 159
 - 7.2.1 Finding the Rank of a Matrix, 159
 - 7.2.2 Calculating the Inverse of a Matrix, 160
 - 7.2.3 Converting to a Triangular Matrix, 161
 - 7.2.4 Calculating the Determinant of a Matrix, 162
 - 7.2.5 Solving a System of Linear Equations, 162
- 7.3 The Dual Linear Program, 165
 - 7.3.1 The Linear Program in Standard Form, 166
 - 7.3.2 Formulating the Dual Problem, 167
 - 7.3.3 Economic Interpretation of the Dual, 170
 - 7.3.4 Importance of the Dual, 171
- 7.4 Relationships Between Primal and Dual Solutions, 171
 - 7.4.1 Relationships Between All Primal and All Dual Feasible Solutions, 171
 - 7.4.2 Relationship Between Primal and Dual Optimum Solutions, 172

- 7.4.3 Relationships Between Each Complementary Pair of Variables at Optimum, 173
- 7.5 Notes, 175
- 7.6 Exercises, 176

8 Linear Programming: Geometric Concepts

180

- 8.1 Geometric Solution, 180
 - 8.1.1 Objective Function, 181
 - 8.1.2 Solution Space, 181
 - 8.1.3 Requirement Space, 183
- 8.2 Convex Sets, 188
 - 8.2.1 Convex Sets and Polyhedra, 188
 - 8.2.2 Directions of Unbounded Convex Sets, 191
 - 8.2.3 Convex and Polyhedral Cones, 191
 - 8.2.4 Convex and Concave Functions, 192
- 8.3 Describing a Bounded Polyhedron, 194
 - 8.3.1 Representation by Extreme Points, 194
 - 8.3.2 Example Application of Representation Theorem, 194
- 8.4 Describing Unbounded Polyhedron, 195
 - 8.4.1 Finding Extreme Direction Algebraically, 195
 - 8.4.2 Representing by Extreme Points and Extreme Directions, 199
 - 8.4.3 Example of Representation Theorem, 199
- 8.5 Faces, Facets, and Dimension of a Polyhedron, 199
- 8.6 Describing a Polyhedron by Facets, 201
- 8.7 Correspondence Between Algebraic and Geometric Terms, 202
- 8.8 Notes, 203
- 8.9 Exercises, 203

9 Linear Programming: Solution Methods

207

- 9.1 Linear Programs in Canonical Form, 207
- 9.2 Basic Feasible Solutions and Reduced Costs, 209
 - 9.2.1 Basic Feasible Solution, 209
 - 9.2.2 Adjacent Basic Feasible Solution, 211
 - 9.2.3 Reduced Costs, 212
- 9.3 The Simplex Method, 213
 - 9.3.1 Better and Feasible Solution, 213
 - 9.3.2 Updating Simplex Tableau by Pivoting, 215
 - 9.3.3 Optimality Test, 216
 - 9.3.4 *Initial Basic Feasible Solution*, 216
- 9.4 Interpreting the Simplex Tableau, 218
 - 9.4.1 Entire Simplex Tableau, 218
 - 9.4.2 Rows of Simplex Tableau, 218
 - 9.4.3 Columns of Simplex Tableau, 219
 - 9.4.4 Pivot Column and Pivot Row, 219

9.4.5	Predicting the New Objective Value Before Updating,	219
9.5	Geometric Interpretation of the Simplex Method,	220
9.5.1	Basic Feasible Solution Versus Extreme Point,	220
9.5.2	Explanation of “Simplex Method” Nomenclature,	222
9.5.3	Identifying an Extreme Ray in a Simplex Tableau,	223
9.6	The Simplex Method for Upper Bounded Variables,	227
9.7	The Dual Simplex Method,	231
9.8	The Revised Simplex Method,	233
9.9	Notes,	239
9.10	Exercises,	240
10	Network Optimization Problems and Solutions	246
10.1	Network Fundamentals,	247
10.2	A Class of Easy Network Problems,	248
10.2.1	The Minimum Cost Network Flow Problem,	249
10.2.2	Formulating the Transportation–Assignment Problem as an MCNF Problem,	249
10.2.3	Formulating the Transshipment Problem as an MCNF Problem,	251
10.2.4	Formulating the Maximum Flow Problem as an MCNF Problem,	251
10.2.5	Formulating the Shortest Path Problem as an MCNF Problem,	251
10.3	Totally Unimodular Matrices,	252
10.3.1	Definition,	252
10.3.2	Sufficient Condition for a Totally Unimodular Matrix,	252
10.3.3	Some Properties of Totally Unimodular Matrices,	254
10.3.4	Matrix Structure of the MCNF Problem,	254
10.3.5	Lower Triangular Matrix and Forward Substitution,	255
10.3.6	Naturally Integer Solution for the MCNF Problem,	255
10.4	The Network Simplex Method,	256
10.4.1	Feasible Spanning Trees Versus Basic Feasible Solutions,	256
10.4.2	The Network Algorithm,	257
10.4.3	Numerical Example,	258
10.5	Solution via LINGO,	264
10.6	Notes,	264
10.7	Exercises,	265
PART III	SOLUTIONS	269
11	Classical Solution Approaches	271
11.1	Branch-and-Bound Approach,	272
11.1.1	Basic Concepts,	272
11.1.2	Branch-and-Bound Algorithm,	278

- 11.2 Cutting Plane Approach, 280
 - 11.2.1 Dual Cutting Plane Approach, 280
 - 11.2.2 Fractional Cutting Plane Method, 281
 - 11.2.3 Mixed Integer Cutting Plane Method, 285
- 11.3 Group Theoretic Approach, 286
 - 11.3.1 Group Theory Terminology, 287
 - 11.3.2 Deriving the Group (Minimization) Problem, 288
 - 11.3.3 Formulating a Group Problem, 290
 - 11.3.4 Solving Group Problem as a Shortest Route Problem, 291
 - 11.3.5 Solving the Original Integer Program, 293
- 11.4 Geometric Concepts, 294
 - 11.4.1 Various Polyhedrons in Original Space, 295
 - 11.4.2 Corner Polyhedron in Solution Space of Nonbasic Variables, 297
- 11.5 Notes, 299
- 11.6 Exercises, 300

12 Branch-and-Cut Approach

305

- 12.1 Introduction, 306
 - 12.1.1 Basic Concept, 306
 - 12.1.2 Branch-and-Cut Algorithm, 306
 - 12.1.3 Generating Valid Cuts and Preprocessing, 307
- 12.2 Valid Inequalities, 308
 - 12.2.1 Valid Inequalities for Linear Programs, 308
 - 12.2.2 Valid Inequalities for Integer Programs, 308
 - 12.2.3 Types of Valid Inequalities, 308
- 12.3 Cut Generating Techniques, 309
 - 12.3.1 Rounding Technique, 310
 - 12.3.2 Disjunction Technique, 310
 - 12.3.3 Lifting Technique, 312
- 12.4 Cuts Generated from Sets Involving Pure Integer Variables, 313
 - 12.4.1 Gomory Fractional Cut, 313
 - 12.4.2 Chvátal–Gomory Cut, 313
 - 12.4.3 Pure Integer Rounding Cut, 314
 - 12.4.4 Objective Integrality Cut, 315
- 12.5 Cuts Generated from Sets Involving Mixed Integer Variables, 315
 - 12.5.1 Gomory Mixed Integer Cut, 315
 - 12.5.2 Mixed Integer Rounding Cut, 319
- 12.6 Cuts Generated from 0–1 Knapsack Sets, 320
 - 12.6.1 Knapsack Cover, 320
 - 12.6.2 Lifted Knapsack Cover, 321
 - 12.6.3 GUB Cover, 323
- 12.7 Cuts Generated from Sets Containing 0–1 Coefficients and 0–1 Variables, 324

12.8	Cuts Generated from Sets with Special Structures, 326	
12.8.1	Flow Cover from Fixed-Charge Flow Network, 326	
12.8.2	Plant/Facility Location (Fixed-Charge Transportation), 327	
12.9	Notes, 329	
12.10	Exercises, 330	
13	Branch-and-Price Approach	334
13.1	Concepts of Branch-and-Price, 334	
13.2	Dantzig–Wolfe Decomposition, 335	
13.3	Generalized Assignment Problem, 344	
13.3.1	Conventional Formulation, 345	
13.3.2	Column Generation Formulation, 345	
13.3.3	Initial Solution, 348	
13.4	GAP Example, 348	
13.4.1	GAP Branching Scheme, 353	
13.4.2	Tailing-Off Effect of Column Generation, 353	
13.4.3	Treatment of Identical Machines, 354	
13.4.4	Branch-and-Price Algorithm, 356	
13.5	Other Application Areas, 356	
13.6	Notes, 357	
13.7	Exercises, 358	
14	Solution via Heuristics, Relaxations, and Partitioning	359
14.1	Introduction, 359	
14.2	Overall Solution Strategy, 359	
14.2.1	Better Formulation by Preprocessing, 360	
14.2.2	LP-Based Branch-and-Bound Framework, 361	
14.2.3	Heuristics for Tightening Lower Bounds, 361	
14.2.4	Relaxations for Tightening Upper Bounds, 362	
14.2.5	Strong Cuts for Tightening Solution Polyhedron, 362	
14.3	Primal Solution via Heuristics, 363	
14.3.1	Local Search Approaches, 364	
14.3.2	Artificial Intelligence Approaches, 366	
14.4	Dual Solution via Relaxation, 373	
14.4.1	Linear Programming Relaxation, 373	
14.4.2	Combinatorial Relaxation, 374	
14.4.3	Lagrangian Relaxation, 376	
14.5	Lagrangian Dual, 377	
14.5.1	Lagrangian Dual in LP, 378	
14.5.2	Lagrangian Dual in IP, 378	
14.5.3	Properties of the Lagrangian Dual, 379	
14.6	Primal–Dual Solution via Benders’ Partitioning, 380	
14.7	Notes, 383	
14.8	Exercises, 383	

- 15.1 Introduction, 387
- 15.2 Typical IP Software Components, 388
 - 15.2.1 Solvers, 388
 - 15.2.2 Presolvers, 389
 - 15.2.3 Modeling Languages, 389
 - 15.2.4 User's Options/Intervention, 390
 - 15.2.5 Data and Application Interfaces, 391
- 15.3 The AMPL Modeling Language, 392
 - 15.3.1 Components of the AMPL Modeling Language, 392
 - 15.3.2 An AMPL Example: the Diet Problem, 393
 - 15.3.3 Enhanced AMPL Modeling Techniques, 397
 - 15.3.4 AMPL Compatible MIP Solvers, 400
- 15.4 LINGO Modeling Language, 400
 - 15.4.1 Prescription of Tolerances, 401
 - 15.4.2 Presolver—Automatic Problem Reduction, 402
 - 15.4.3 Solvers for Linear/Integer Programming, 402
 - 15.4.4 Interfacing with the User, 403
 - 15.4.5 LINGO Modeling Conventions, 403
 - 15.4.6 LINGO Model for the Diet Problem, 404
- 15.5 MPL Modeling Language, 405
 - 15.5.1 MPL Modeling Conventions, 406
 - 15.5.2 MPL Model for the Diet Problem, 408
 - 15.5.3 MPL Compatible MIP Solvers, 409