Graduate Texts in Mathematics 271

Graduate Texts in Mathematics

Graduate Texts in Mathematics bridge the gap between passive study and creative understanding, offering graduate-level introductions to advanced topics in mathematics. The volumes are carefully written as teaching aids and highlight characteristic features of the theory. Although these books are frequently used as textbooks in graduate courses, they are also suitable for individual study.

More information about this series at http://www.springer.com/series/136

Michele Conforti • Gérard Cornuéjols
Giacomo Zambelli

Integer Programming

Springer

Michele Conforti
Department of Mathematics
University of Padova
Padova, Italy

Gérard Cornuéjols
Tepper School of Business
Carnegie Mellon University
Pittsburgh, PA, USA

Giacomo Zambelli
Department of Management
London School of Economics
and Political Science
London, UK

ISSN 0072-5285 ISSN 2197-5612 (electronic)
ISBN 978-3-319-11007-3 ISBN 978-3-319-11008-0 (eBook)
DOI 10.1007/978-3-319-11008-0
Springer Cham Heidelberg New York Dordrecht London

Library of Congress Control Number: 2014952029

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Preface

Integer programming is a thriving area of optimization, which is applied nowadays to a multitude of human endeavors, thanks to high quality software. It was developed over several decades and is still evolving rapidly.

The goal of this book is to present the mathematical foundations of integer programming, with emphasis on the techniques that are most successful in current software implementations: convexification and enumeration.

This textbook is intended for a graduate course in integer programming in M.S. or Ph.D. programs in applied mathematics, operations research, industrial engineering, or computer science.

To better understand the excitement that is generated today by this area of mathematics, it is helpful to provide a historical perspective.

Babylonian tablets show that mathematicians were already solving systems of linear equations over 3,000 years ago. The eighth book of the Chinese *Nine Books of Arithmetic*, written over 2,000 years ago, describes what is now known as the Gaussian elimination method. In 1809, Gauss [160] used this method in his work, stating that it was a "standard technique." The method was subsequently named after him.

A major breakthrough occurred when mathematicians started analyzing systems of linear *inequalities*. This is a fertile ground for beautiful theories. In 1826 Fourier [145] gave an algorithm for solving such systems by eliminating variables one at a time. Other important contributions are due to Farkas [135] and Minkowski [279]. Systems of linear inequalities define *polyhedra* and it is natural to optimize a linear function over them. This is the topic of *linear programming*, arguably one of the greatest successes of computational mathematics in the twentieth century. The *simplex method*, developed by Dantzig [102] in 1951, is currently used to solve large-scale problems in all sorts of application areas. It is often desirable to find *integer solutions* to linear programs. This is the topic of this book. The first algorithm for solving pure integer linear programs was discovered in 1958 by Gomory [175].

When considering algorithmic questions, a fundamental issue is the increase in computing time when the size of the problem instance increases. In the 1960s Edmonds [123] was one of the pioneers in stressing the importance of *polynomial-time* algorithms. These are algorithms whose computing time is bounded by a polynomial function of the instance size. In particular Edmonds [125] pointed out that the Gaussian elimination method can be turned into a polynomial-time algorithm by being a bit careful with the intermediate numbers that are generated. The existence of a polynomial-time algorithm for linear programming remained a challenge for many years. This question was resolved positively by Khachiyan [235] in 1979, and later by Karmarkar [229] using a totally different algorithm. Both algorithms were (and still are) very influential, each in its own way. In integer programming, Lenstra [256] found a polynomial-time algorithm when the number of variables is fixed.

Although integer programming is NP-hard in general, the *polyhedral approach* has proven successful in practice. It can be traced back to the work of Dantzig, Fulkerson, and Johnson [103] in 1954. Research is currently very active in this area. Beautiful mathematical results related to the polyhedral approach pervade the area of integer programming. This book presents several of these results. Also very promising are nonpolyhedral approximations that can be computed in polynomial-time, such as semidefinite relaxations, see Lovász and Schrijver [264], and Goemans and Williamson [173].

We are grateful to the colleagues and students who read earlier drafts of this book and have helped us improve it. In particular many thanks to Lawrence Wolsey for carefully checking the whole manuscript. Many thanks also to Marco Di Summa, Kanstantsin Pashkovich, Teresa Provesan, Sercan Yildiz, Monique Laurent, Sebastian Pokutta, Dan Bienstock, François Margot, Giacomo Nannicini, Juan Pablo Viema, Babis Tsourakakis, Thiago Serra, Yang Jiao, and Tarek Elgindy for their excellent suggestions.

Thank you to Julie Zavon for the artwork at the end of most chapters.

This work was supported in part by NSF grant CMMI1263239 and ONR grant N000141210032.

Padova, Italy — Michele Conforti
Pittsburgh, PA, USA — Gérard Cornuéjols
London, UK — Giacomo Zambelli

Contents

1 Getting Started **1**
1.1 Integer Programming 1
1.2 Methods for Solving Integer Programs 5
1.2.1 The Branch-and-Bound Method 6
1.2.2 The Cutting Plane Method 11
1.2.3 The Branch-and-Cut Method 15
1.3 Complexity 16
1.3.1 Problems, Instances, Encoding Size 17
1.3.2 Polynomial Algorithm 18
1.3.3 Complexity Class NP 19
1.4 Convex Hulls and Perfect Formulations 20
1.4.1 Example: A Two-Dimensional Mixed Integer Set . . . 22
1.4.2 Example: A Two-Dimensional Pure Integer Set 24
1.5 Connections to Number Theory 25
1.5.1 The Greatest Common Divisor 26
1.5.2 Integral Solutions to Systems of Linear Equations . . 29
1.6 Further Readings 36
1.7 Exercises 38

2 Integer Programming Models **45**
2.1 The Knapsack Problem 45
2.2 Comparing Formulations 46
2.3 Cutting Stock: Formulations with Many Variables 48
2.4 Packing, Covering, Partitioning 51
2.4.1 Set Packing and Stable Sets 51
2.4.2 Strengthening Set Packing Formulations 52
2.4.3 Set Covering and Transversals 53
2.4.4 Set Covering on Graphs: Many Constraints 55

2.4.5 Set Covering with Many Variables: Crew Scheduling . 57
2.4.6 Covering Steiner Triples 58
2.5 Generalized Set Covering: The Satisfiability Problem 58
2.6 The Sudoku Game 60
2.7 The Traveling Salesman Problem 61
2.8 The Generalized Assignment Problem 65
2.9 The Mixing Set 66
2.10 Modeling Fixed Charges 66
2.10.1 Facility Location 67
2.10.2 Network Design 69
2.11 Modeling Disjunctions 70
2.12 The Quadratic Assignment Problem
and Fortet's Linearization 72
2.13 Further Readings 73
2.14 Exercises 74

3 Linear Inequalities and Polyhedra 85
3.1 Fourier Elimination 85
3.2 Farkas' Lemma 88
3.3 Linear Programming 89
3.4 Affine, Convex, and Conic Combinations 91
3.4.1 Linear Combinations, Linear Spaces 91
3.4.2 Affine Combinations, Affine Spaces 92
3.4.3 Convex Combinations, Convex Sets 92
3.4.4 Conic Combinations, Convex Cones 93
3.5 Polyhedra and the Theorem of Minkowski–Weyl 94
3.5.1 Minkowski–Weyl Theorem for Polyhedral Cones . . . 94
3.5.2 Minkowski–Weyl Theorem for Polyhedra 95
3.6 Lineality Space and Recession Cone 97
3.7 Implicit Equalities, Affine Hull, and Dimension 98
3.8 Faces 101
3.9 Minimal Representation and Facets 104
3.10 Minimal Faces 108
3.11 Edges and Extreme Rays 109
3.12 Decomposition Theorem for Polyhedra 111
3.13 Encoding Size of Vertices, Extreme Rays, and Facets 112
3.14 Carathéodory's Theorem 113
3.15 Projections 116
3.16 Polarity 119

3.17 Further Readings . . . 120
3.18 Exercises . . . 124

4 Perfect Formulations 129
4.1 Properties of Integral Polyhedra . . . 130
4.2 Total Unimodularity . . . 131
4.3 Networks . . . 134
4.3.1 Circulations . . . 135
4.3.2 Shortest Paths . . . 137
4.3.3 Maximum Flow and Minimum Cut . . . 139
4.4 Matchings in Graphs . . . 145
4.4.1 Augmenting Paths . . . 146
4.4.2 Cardinality Bipartite Matchings . . . 147
4.4.3 Minimum Weight Perfect Matchings in Bipartite Graphs . . . 149
4.4.4 The Matching Polytope . . . 150
4.5 Spanning Trees . . . 153
4.6 Total Dual Integrality . . . 155
4.7 Submodular Polyhedra . . . 157
4.8 The Fundamental Theorem of Integer Programming . . . 159
4.8.1 An Example: The Mixing Set . . . 161
4.8.2 Mixed Integer Linear Programming is in NP . . . 163
4.8.3 Polynomial Encoding of the Facets of the Integer Hull 165
4.9 Union of Polyhedra . . . 166
4.9.1 Example: Modeling Disjunctions . . . 169
4.9.2 Example: All the Even Subsets of a Set . . . 170
4.9.3 Mixed Integer Linear Representability . . . 171
4.10 The Size of a Smallest Perfect Formulation . . . 174
4.10.1 Rectangle Covering Bound . . . 177
4.10.2 An Exponential Lower-Bound for the Cut Polytope . . 179
4.10.3 An Exponential Lower-Bound for the Matching Polytope . . . 181
4.11 Further Readings . . . 182
4.12 Exercises . . . 187

5 Split and Gomory Inequalities 195
5.1 Split Inequalities . . . 195
5.1.1 Inequality Description of the Split Closure . . . 199
5.1.2 Polyhedrality of the Split Closure . . . 202
5.1.3 Split Rank . . . 203

5.1.4 Gomory's Mixed Integer Inequalities 205
5.1.5 Mixed Integer Rounding Inequalities 206
5.2 Chvátal Inequalities 207
5.2.1 The Chvátal Closure of a Pure Integer Linear Set . . . 208
5.2.2 Chvátal Rank 209
5.2.3 Chvátal Inequalities for Other Forms of the Linear System 211
5.2.4 Gomory's Fractional Cuts 212
5.2.5 Gomory's Lexicographic Method for Pure Integer Programs 213
5.3 Gomory's Mixed Integer Cuts 216
5.4 Lift-and-Project 222
5.4.1 Lift-and-Project Rank for Mixed 0,1 Linear Programs 223
5.4.2 A Finite Cutting Plane Algorithm for Mixed 0, 1 Linear Programming 225
5.5 Further Readings 227
5.6 Exercises 230

6 Intersection Cuts and Corner Polyhedra 235
6.1 Corner Polyhedron 235
6.2 Intersection Cuts 240
6.2.1 The Gauge Function 248
6.2.2 Maximal Lattice-Free Convex Sets 249
6.3 Infinite Relaxations 253
6.3.1 Pure Integer Infinite Relaxation 256
6.3.2 Continuous Infinite Relaxation 264
6.3.3 The Mixed Integer Infinite Relaxation 268
6.3.4 Trivial and Unique Liftings 270
6.4 Further Readings 273
6.5 Exercises 275

7 Valid Inequalities for Structured Integer Programs 281
7.1 Cover Inequalities for the 0,1 Knapsack Problem 282
7.2 Lifting 283
7.2.1 Lifting Minimal Cover Inequalities 285
7.2.2 Lifting Functions, Superadditivity, and Sequence Independent Lifting 287
7.2.3 Sequence Independent Lifting for Minimal Cover Inequalities 289

7.3 Flow Cover Inequalities 291
7.4 Faces of the Symmetric Traveling Salesman Polytope 299
7.4.1 Separation of Subtour Elimination Constraints 302
7.4.2 Comb Inequalities 303
7.4.3 Local Cuts 305
7.5 Equivalence Between Optimization and Separation 307
7.6 Further Readings 311
7.7 Exercises 315

8 Reformulations and Relaxations 321
8.1 Lagrangian Relaxation 321
8.1.1 Examples 324
8.1.2 Subgradient Algorithm 326
8.2 Dantzig–Wolfe Reformulation 330
8.2.1 Problems with Block Diagonal Structure 332
8.2.2 Column Generation 334
8.2.3 Branch-and-Price 337
8.3 Benders Decomposition 338
8.4 Further Readings 341
8.5 Exercises 344

9 Enumeration 351
9.1 Integer Programming in Fixed Dimension 351
9.1.1 Basis Reduction 352
9.1.2 The Flatness Theorem and Rounding Polytopes . . . 358
9.1.3 Lenstra's Algorithm 362
9.2 Implementing Branch-and-Cut 364
9.3 Dealing with Symmetries 373
9.4 Further Readings 380
9.5 Exercises 383

10 Semidefinite Bounds 389
10.1 Semidefinite Relaxations 389
10.2 Two Applications in Combinatorial Optimization 391
10.2.1 The Max-Cut Problem 391
10.2.2 The Stable Set Problem 393
10.3 The Lovász–Schrijver Relaxation 394
10.3.1 Semidefinite Versus Linear Relaxations 396

10.3.2 Connection with Lift-and-Project 397
10.3.3 Iterating the Lovász–Schrijver Procedure 399
10.4 The Sherali–Adams and Lasserre Hierarchies 400
10.4.1 The Sherali–Adams Hierarchy 400
10.4.2 The Lasserre Hierarchy 402
10.5 Further Readings 408
10.6 Exercises 411

Bibliography **415**

Index **447**

The authors discussing the outline of the book