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

Springer

http://www.springer.com/978-3-319-11007-3

Integer Programming
Conforti, M.; Cornuejols, G.; Zambelli, G.
2014, XII, 456 p. 75 illus., Hardcover
ISBN: 978-3-319-11007-3