Preface

An approximation algorithm is an efficient algorithm that produces solutions to an optimization problem that are guaranteed to be within a fixed ratio of the optimal solution. Instead of spending an exponential amount of time finding the optimal solution, an approximation algorithm settles for near-optimal solutions within polynomial time in the input size. Approximation algorithms have been studied since the mid-1960s. Their importance was, however, not fully understood until the discovery of the **NP**-completeness theory. Many well-known optimization problems have been proved, under reasonable assumptions in this theory, to be intractable, in the sense that optimal solutions to these problems are not computable within polynomial time. As a consequence, near-optimal approximation algorithms are the best one can expect when trying to solve these problems.

In the past decade, the area of approximation algorithms has experienced an explosive rate of growth. This growth rate is partly due to the development of related research areas, such as data mining, communication networks, bioinformatics, and computational game theory. These newly established research areas generate a large number of new, intractable optimization problems, most of which have direct applications to real-world problems, and so efficient approximate solutions to them are actively sought after.

In addition to the external, practical need for efficient approximation algorithms, there is also an intrinsic, theoretical motive behind the research of approximation algorithms. In the design of an exact-solution algorithm, the main, and often only, measure of the algorithm's performance is its running time. This fixed measure often limits our choice of techniques in the algorithm's design. For an approximation algorithm, however, there is an equally important second measure, that is, the performance ratio of the algorithm, which measures how close the approximation algorithm's output is to the optimal solution. This measure adds a new dimension to the design and analysis of approximation algorithms. Namely, we can now study the tradeoff between the running time and the performance ratio of approximation algorithms, and apply different design techniques to achieve different tradeoffs between these two measures. In addition, new theoretical issues about the approximation to an optimization problem need to be addressed: What is the performance ratio of an approximation algorithm for this problem based on certain types of design strategy? What is the best performance ratio of any polynomial-time approximation algorithm for this problem? Does the problem have a polynomial-time approximation scheme or a fully polynomial-time approximation scheme? These questions are not only of significance in practice for the design of approximation algorithms; they are also of great theoretical interest, with intriguing connections to the **NP**-completeness theory.

Motivated by these theoretical questions and the great number of newly discovered optimization problems, people have developed many new design techniques for approximation algorithms, including the greedy strategy, the restriction method, the relaxation method, partition, local search, power graphs, and linear and semidefinite programming. A comprehensive survey of all these methods and results in a single book is not possible. We instead provide in this book an intensive study of the main methods, with abundant applications following our discussion of each method. Indeed, this book is organized according to design methods instead of application problems. Thus, one can study approximation algorithms of the same nature together, and learn about the design techniques in a more unified way. To this end, the book is arranged in the following way: First, in Chapter 1, we give a brief introduction to the concept of **NP**-completeness and approximation algorithms. In Chapter 2, we give an in-depth analysis of the greedy strategy, including greedy algorithms with submodular potential functions and those with nonsubmodular potential functions. In Chapters 3, 4, and 5, we cover various restriction methods, including partition and Guillotine cut methods, with applications to many geometric problems. In the next four chapters, we study the relaxation methods. In addition to a general discussion of the relaxation method in Chapter 6, we devote three chapters to approximation algorithms based on linear and semidefinite programming, including the primal-dual schema and its equivalence with the local ratio method. Finally, in Chapter 10, we present various inapproximability results based on recent work in the **NP**-completeness theory. A number of examples and exercises are provided for each design technique. They are drawn from diverse areas of research, including communication network design, optical networks, wireless ad hoc networks, sensor networks, bioinformatics, social networks, industrial engineering, and information management systems.

This book has grown out of lecture notes used by the authors at the University of Minnesota, University of Texas at Dallas, Tsinghua University, Graduate School of Chinese Academy of Sciences, Xi'an Jiaotong University, Zhejiang University, East China Normal University, Dalian University of Technology, Xinjiang University, Nankai University, Lanzhou Jiaotong University, Xidian University, and Harbin Institute of Technology. In a typical one-semester class for first-year graduate students, one may cover the first two chapters, one or two chapters on the restriction method, two or three chapters on the relaxation method, and Chapter 10. With more advanced students, one may also teach a seminar course focusing on one of the greedy, restriction, or relaxation methods, based on the corresponding chapters of this book and supplementary material from recent research papers. For instance, a seminar on combinatorial optimization emphasizing approximations based on linear and semidefinite programming can be organized using Chapters 7, 8, and 9.

This book has benefited much from the help of our friends, colleagues, and students. We are indebted to Peng-Jun Wan, Weili Wu, Xiuzhen Cheng, Jie Wang, Yinfeng Xu, Zhao Zhang, Deying Li, Hejiao Huang, Hong Zhu, Guochuan Zhang, Wei Wang, Shugang Gao, Xiaofeng Gao, Feng Zou, Ling Ding, Xianyue Li, My T. Thai, Donghyun Kim, J. K. Willson, and Roozbeh Ebrahimi Soorchaei, who made much-valued suggestions and corrections to the earlier drafts of the book. We are also grateful to Professors Frances Yao, Richard Karp, Ronald Graham, and Fan Chung for their encouragement. Special thanks are due to Professor Andrew Yao and the Institute for Theoretical Computer Science, Tsinghua University, for the generous support and stimulating environment they provided for the first two authors during their numerous visits to Tsinghua University.

Dallas, Texas Ding-Zhu Du
Stony Brook, New York Ker-I Ko
Beijing, China Xiaodong Hu
August 2011

Springer

http://www.springer.com/978-1-4614-1700-2

Design and Analysis of Approximation Algorithms
Du, D.-Z.; Ko, K.-I.; Hu, X.
2012, XII, 440 p., Hardcover
ISBN: 978-1-4614-1700-2