A Gentle Introduction to Optimization

Optimization is an essential technique for solving problems in areas as diverse as accounting, computer science and engineering. Assuming only basic linear algebra and with a clear focus on the fundamental concepts, this textbook is the perfect starting point for first- and second-year undergraduate students from a wide range of backgrounds and with varying levels of ability.

- Modern, real-world examples motivate the theory throughout.
- Over 140 exercises, ranging from the routine to the more advanced, give readers the opportunity to try out the skills they gain in each section.
- Solutions are available for instructors as well as algorithms for computational problems.
- Self-contained chapters allow instructors and students to tailor the material to their own needs and make the book suitable for self-study.
- Suggestions for further reading help students to take the next step to more advanced courses in optimization.
- Material has been thoroughly tried and tested by the authors, who together have 40 years of teaching experience.

B. Guenin is Professor in the Department of Combinatorics and Optimization at the University of Waterloo. He received a Fulkerson Prize awarded jointly by the Mathematical Programming Society and the American Mathematical Society in 2003. He is also the recipient of a Premier’s Research Excellence Award in 2001 from the Government of Ontario, Canada. Guenin currently serves on the Editorial Board of the SIAM Journal of Discrete Mathematics.

J. Könemann is Professor in the Department of Combinatorics and Optimization at the University of Waterloo. He received an IBM Corporation Faculty Award in 2005, and an Early Researcher Award from the Government of Ontario, Canada, in 2007. He served on the program committees of several major conferences in Mathematical Optimization and Computer Science, and is a member of the editorial board of Elsevier’s Surveys in Operations Research and Management Science.

L. Tuncel is Professor in the Department of Combinatorics and Optimization at the University of Waterloo. In 1999 he received a Premier’s Research Excellence Award from the Government of Ontario, Canada. More recently, he received a Faculty of Mathematics Award for Distinction in Teaching from the University of Waterloo in 2012. Tunçel currently serves on the Editorial Board of the SIAM Journal on Optimization and as an Associate Editor of Mathematics of Operations Research.
A Gentle Introduction to Optimization

B. GUENIN
J. KÖNEMANN
L. TUNÇEL

University of Waterloo, Ontario
Contents

Preface ix

1 Introduction 1
  1.1 A first example 2
    1.1.1 The formulation 3
    1.1.2 Correctness 4
  1.2 Linear programs 5
    1.2.1 Multi-period models 7
  1.3 Integer programs 14
    1.3.1 Assignment problem 15
    1.3.2 Knapsack problem 17
  1.4 Optimization problems on graphs 26
    1.4.1 Shortest path problem 27
    1.4.2 Minimum cost perfect matching 28
  1.5 Integer programs continued 30
    1.5.1 Minimum cost perfect matching 30
    1.5.2 Shortest path problem 32
  1.6 Nonlinear programs 37
    1.6.1 Pricing a tech gadget 37
    1.6.2 Finding a closest point feasible in an LP 39
    1.6.3 Finding a “central” feasible solution of an LP 39
  1.7 Overview of the book 42
  1.8 Further reading and notes 43

2 Solving linear programs 44
  2.1 Possible outcomes 44
    2.1.1 Infeasible linear programs 45
    2.1.2 Unbounded linear programs 47
    2.1.3 Linear programs with optimal solutions 48
  2.2 Standard equality form 50
  2.3 A simplex iteration 55
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.4</td>
<td>Bases and canonical forms</td>
<td>57</td>
</tr>
<tr>
<td>2.4.1</td>
<td>Bases</td>
<td>57</td>
</tr>
<tr>
<td>2.4.2</td>
<td>Canonical forms</td>
<td>58</td>
</tr>
<tr>
<td>2.5</td>
<td>The simplex algorithm</td>
<td>64</td>
</tr>
<tr>
<td>2.5.1</td>
<td>An example with an optimal solution</td>
<td>64</td>
</tr>
<tr>
<td>2.5.2</td>
<td>An unbounded example</td>
<td>67</td>
</tr>
<tr>
<td>2.5.3</td>
<td>Formalizing the procedure</td>
<td>68</td>
</tr>
<tr>
<td>2.6</td>
<td>Finding feasible solutions</td>
<td>75</td>
</tr>
<tr>
<td>2.6.1</td>
<td>General scheme</td>
<td>75</td>
</tr>
<tr>
<td>2.6.2</td>
<td>The two phase simplex algorithm—an example</td>
<td>78</td>
</tr>
<tr>
<td>2.6.3</td>
<td>Consequences</td>
<td>80</td>
</tr>
<tr>
<td>2.7</td>
<td>Simplex via tableaus*</td>
<td>82</td>
</tr>
<tr>
<td>2.7.1</td>
<td>Pivoting</td>
<td>82</td>
</tr>
<tr>
<td>2.7.2</td>
<td>Tableaus</td>
<td>84</td>
</tr>
<tr>
<td>2.8</td>
<td>Geometry</td>
<td>85</td>
</tr>
<tr>
<td>2.8.1</td>
<td>Feasible region of LPs and polyhedra</td>
<td>86</td>
</tr>
<tr>
<td>2.8.2</td>
<td>Convexity</td>
<td>88</td>
</tr>
<tr>
<td>2.8.3</td>
<td>Extreme points</td>
<td>89</td>
</tr>
<tr>
<td>2.8.4</td>
<td>Geometric interpretation of the simplex algorithm</td>
<td>92</td>
</tr>
<tr>
<td>2.9</td>
<td>Further reading and notes</td>
<td>97</td>
</tr>
<tr>
<td>3</td>
<td>Duality through examples</td>
<td>99</td>
</tr>
<tr>
<td>3.1</td>
<td>The shortest path problem</td>
<td>99</td>
</tr>
<tr>
<td>3.1.1</td>
<td>An intuitive lower bound</td>
<td>100</td>
</tr>
<tr>
<td>3.1.2</td>
<td>A general argument – weak duality</td>
<td>103</td>
</tr>
<tr>
<td>3.1.3</td>
<td>Revisiting the intuitive lower bound</td>
<td>106</td>
</tr>
<tr>
<td>3.1.4</td>
<td>An algorithm</td>
<td>111</td>
</tr>
<tr>
<td>3.1.5</td>
<td>Correctness of the algorithm</td>
<td>114</td>
</tr>
<tr>
<td>3.2</td>
<td>Minimum cost perfect matching in bipartite graphs</td>
<td>116</td>
</tr>
<tr>
<td>3.2.1</td>
<td>An intuitive lower bound</td>
<td>117</td>
</tr>
<tr>
<td>3.2.2</td>
<td>A general argument–weak duality</td>
<td>119</td>
</tr>
<tr>
<td>3.2.3</td>
<td>Revisiting the intuitive lower bound</td>
<td>122</td>
</tr>
<tr>
<td>3.2.4</td>
<td>An algorithm</td>
<td>125</td>
</tr>
<tr>
<td>3.2.5</td>
<td>Correctness of the algorithm</td>
<td>129</td>
</tr>
<tr>
<td>3.2.6</td>
<td>Finding perfect matchings in bipartite graphs*</td>
<td>133</td>
</tr>
<tr>
<td>3.3</td>
<td>Further reading and notes</td>
<td>140</td>
</tr>
<tr>
<td>4</td>
<td>Duality theory</td>
<td>142</td>
</tr>
<tr>
<td>4.1</td>
<td>Weak duality</td>
<td>142</td>
</tr>
<tr>
<td>4.2</td>
<td>Strong duality</td>
<td>150</td>
</tr>
<tr>
<td>4.3</td>
<td>A geometric characterization of optimality</td>
<td>153</td>
</tr>
<tr>
<td>4.3.1</td>
<td>Complementary slackness</td>
<td>154</td>
</tr>
<tr>
<td>4.3.2</td>
<td>Geometry</td>
<td>158</td>
</tr>
</tbody>
</table>
4.4 Farkas’ lemma* 163
4.5 Further reading and notes 165

5 Applications of duality*
5.1 Approximation algorithm for set-cover 166
  5.1.1 A primal–dual algorithm 167
  5.1.2 Greed is good ... at least sometimes 170
  5.1.3 Discussion 172
5.2 Economic interpretation 173
5.3 The maximum-flow–minimum-cut theorem 176
  5.3.1 Totally unimodular matrices 178
  5.3.2 Applications to st-flows 179

6 Solving integer programs
6.1 Integer programs versus linear programs 183
6.2 Cutting planes 188
  6.2.1 Cutting planes and the simplex algorithm 190
6.3 Branch and bound 196
6.4 Traveling salesman problem and a separation algorithm* 201
6.5 Further reading and notes 204

7 Nonlinear optimization
7.1 Some examples 206
7.2 Some nonlinear programs are very hard
  7.2.1 NLP versus 0,1 integer programming 208
  7.2.2 Hard small-dimensional instances 209
7.3 Convexity
  7.3.1 Convex functions and epigraphs 210
  7.3.2 Level sets and feasible region 213
7.4 Relaxing convex NLPs
  7.4.1 Subgradients 216
  7.4.2 Supporting halfspaces 216
7.5 Optimality conditions for the differentiable case
  7.5.1 Sufficient conditions for optimality 218
  7.5.2 Differentiability and gradients 220
  7.5.3 A Karush–Kuhn–Tucker theorem 221
7.6 Optimality conditions based on Lagrangians 223
7.7 Nonconvex optimization problems
  7.7.1 The Karush–Kuhn–Tucker theorem for nonconvex problems* 227
  7.7.2 Convex relaxation approach to nonconvex problems* 230
7.8 Interior-point method for linear programs* 234
  7.8.1 A polynomial-time interior-point algorithm* 239
7.9 Further reading and notes 242

Appendix A Computational complexity 244
A.1 What is a fast (resp. slow) algorithm? 244
  A.1.1 The big “O” notation 245
  A.1.2 Input size and running time 245
  A.1.3 Polynomial and exponential algorithms 247
A.2 Examples of fast and slow algorithms 249
  A.2.1 Linear programming 249
  A.2.2 Other algorithms in this book 251
A.3 The classes $NP$, $co-NP$ and $P$ 252
  A.3.1 Decision problems 252
  A.3.2 The class $NP$ 253
  A.3.3 The class $co-NP$ 255
  A.3.4 The class $P$ 256
A.4 Hard problems 258
  A.4.1 Reducibility 258
  A.4.2 NP-complete problems 260
  A.4.3 Complexity as guide 261
  A.4.4 Easy versus hard problems 261

References 263
Index 267
Preface

Desire to improve drives many human activities. Optimization can be seen as a means for identifying better solutions by utilizing a scientific and mathematical approach. In addition to its widespread applications, optimization is an amazing subject with very strong connections to many other subjects and deep interactions with many aspects of computation and theory. The main goal of this textbook is to provide an attractive, modern, and accessible route to learning the fundamental ideas in optimization for a large group of students with varying backgrounds and abilities. The only background required for the textbook is a first-year linear algebra course (some readers may even be ready immediately after finishing high school). However, a course based on this book can serve as a header course for all optimization courses. As a result, an important goal is to ensure that the students who successfully complete the course are able to proceed to more advanced optimization courses.

Another goal of ours was to create a textbook that could be used by a large group of instructors, possibly under many different circumstances. To a degree, we tested this over a four-year period. Including the three of us, 12 instructors used the drafts of the book for two different courses. Students in various programs (majors), including accounting, business, software engineering, statistics, actuarial science, operations research, applied mathematics, pure mathematics, computational mathematics, computer science, combinatorics and optimization, have taken these courses. We believe that the book will be suitable for a wide range of students (mathematics, mathematical sciences including computer science, engineering including software engineering, and economics). To accomplish our goals, we operated with the following rules:

1. Always motivate the subject/algorith/item (leading by modern, relatable examples which expose important aspects of the subject/algorith/item).
2. Keep the text as concise and as focused as possible (this meant, that some of the more advanced or tangential topics are either treated in advanced sections or in starred exercises).
3. Make sure that some of the pieces are modular so that an instructor or a reader can choose to skip certain parts of the text smoothly. (Please see the potential usages of the book below.)
In particular, for the derivation and overall presentation of the simplex method, we focused on the main ideas rather than gritty details (which in our opinion and experience, distract from the beauty and power of the method as well as the upcoming generalizations of the underlying ideas).

We emphasized the unifying notion of relaxation in our discussion of duality, integer programming, and combinatorial optimization as well as nonlinear optimization. We also emphasized the power and usefulness of primal–dual approaches as well as convexity in deriving algorithms, understanding the theory, and improving the usage of optimization in applications.

We strived to enhance understanding by weaving in geometric notions, interpretations, and ideas starting with the first chapter, Introduction, and all the way through to the last chapter (Nonlinear optimization) in a cohesive and consistent manner.

We made sure that the themes of efficiency of algorithms and good certificates of correctness as well as their relevance were present. We included a brief introduction to the relevant parts of computational complexity in the appendix.

All of these ideas come to a beautiful meeting point in the last chapter, Nonlinear optimization. First of all, we develop the ideas only based on linear algebraic and geometric notions, capitalizing on the strength built through linear programming (geometry, halfspaces, duality) and discrete optimization (relaxation). We arrive at the powerful Karush–Kuhn–Tucker Theorem without requiring more background in continuous mathematics and real analysis.

We thank Yu Hin (Gary) Au, Joseph Cheriyan, Bill Cook, Bill Cunningham, Ricardo Fukasawa, Konstantinos Georgiou, Stephen New, Clinton Reddekop, Patrick Roh, Laura Sanita and Nick Wormald for very useful suggestions, corrections and ideas for exercises. We also thank the Editor, David Tranah, for very useful suggestions, and for his support and patience.

Some alternative ways of using the book
We designed the textbook so that starred sections/chapters can be skipped without any trouble. In Chapter 3 it is sufficient to pick only one of the two motivating problems (the shortest path or the minimum cost matching problem). Moreover, there are many seamless ways of using the textbook, we outline some of them below.

- For a high-paced, academically demanding course, cover the material from beginning to end by inserting the Appendix (Computational complexity) between Chapter 2 or 3, or 4 or 5.
- Cover in order Chapters 1,2,3,4,5,6,7 (do not cover the Appendix).
- For an audience mostly interested in modeling and applications, cover Chapters 1,2,3,6,7.
- For an audience with prior knowledge of the simplex method, cover Chapters 1,3,4,5,6,7.
- For a slow-paced course based only on linear programming, cover Chapters 1,2,3,4.
- For a course based only on linear programming, cover Chapters 1,2,3,4,5 (possibly with the Appendix included).
• For a course based only on linear programming and discrete optimization, cover chapters 1, 2, 3, 4, 5, 6 (possibly with the Appendix included). This version may be particularly suitable for an introductory course offered in computer science departments.
• For a course based only on linear programming and discrete optimization (but at a slower pace than the last one above), cover Chapters 1, 2, 3, 4, 6.
• For a course based only on linear programming and nonlinear optimization, cover Chapters 1, 2, 3, 4, 5, 7.
• For an audience with some prior course in elementary linear programming, cover Chapters 1, 5, 6, 7 (insert the Appendix after Chapter 5).
• Our book can also be used for independent study and by undergraduate research assistants to quickly build up the required background for research studies.