

**FEASIBILITY AND INFEASIBILITY
IN OPTIMIZATION:
Algorithms and Computational Methods**

**Recent titles in the INTERNATIONAL SERIES IN
OPERATIONS RESEARCH & MANAGEMENT SCIENCE**

Frederick S. Hillier, Series Editor, Stanford University

- Sethi, Yan & Zhang/ *INVENTORY AND SUPPLY CHAIN MANAGEMENT WITH FORECAST UPDATES*
- Cox/ *QUANTITATIVE HEALTH RISK ANALYSIS METHODS: Modeling the Human Health Impacts of Antibiotics Used in Food Animals*
- Ching & Ng/ *MARKOV CHAINS: Models, Algorithms and Applications*
- Li & Sun/ *NONLINEAR INTEGER PROGRAMMING*
- Kaliszewski/ *SOFT COMPUTING FOR COMPLEX MULTIPLE CRITERIA DECISION MAKING*
- Bouyssou et al/ *EVALUATION AND DECISION MODELS WITH MULTIPLE CRITERIA: Stepping stones for the analyst*
- Blecker & Friedrich/ *MASS CUSTOMIZATION: Challenges and Solutions*
- Appa, Pitsoulis & Williams/ *HANDBOOK ON MODELLING FOR DISCRETE OPTIMIZATION*
- Herrmann/ *HANDBOOK OF PRODUCTION SCHEDULING*
- Axsäter/ *INVENTORY CONTROL, 2nd Ed.*
- Hall/ *PATIENT FLOW: Reducing Delay in Healthcare Delivery*
- Józefowska & Węglarz/ *PERSPECTIVES IN MODERN PROJECT SCHEDULING*
- Tian & Zhang/ *VACATION QUEUEING MODELS: Theory and Applications*
- Yan, Yin & Zhang/ *STOCHASTIC PROCESSES, OPTIMIZATION, AND CONTROL THEORY APPLICATIONS IN FINANCIAL ENGINEERING, QUEUEING NETWORKS, AND MANUFACTURING SYSTEMS*
- Saaty & Vargas/ *DECISION MAKING WITH THE ANALYTIC NETWORK PROCESS: Economic, Political, Social & Technological Applications w. Benefits, Opportunities, Costs & Risks*
- Yu/ *TECHNOLOGY PORTFOLIO PLANNING AND MANAGEMENT: Practical Concepts and Tools*
- Kandiller/ *PRINCIPLES OF MATHEMATICS IN OPERATIONS RESEARCH*
- Lee & Lee/ *BUILDING SUPPLY CHAIN EXCELLENCE IN EMERGING ECONOMIES*
- Weintraub/ *MANAGEMENT OF NATURAL RESOURCES: A Handbook of Operations Research Models, Algorithms, and Implementations*
- Hooker/ *INTEGRATED METHODS FOR OPTIMIZATION*
- Dawande et al/ *THROUGHPUT OPTIMIZATION IN ROBOTIC CELLS*
- Friesz/ *NETWORK SCIENCE, NONLINEAR SCIENCE and INFRASTRUCTURE SYSTEMS*
- Cai, Sha & Wong/ *TIME-VARYING NETWORK OPTIMIZATION*
- Mamon & Elliott/ *HIDDEN MARKOV MODELS IN FINANCE*
- del Castillo/ *PROCESS OPTIMIZATION: A Statistical Approach*
- Józefowska/ *JUST-IN-TIME SCHEDULING: Models & Algorithms for Computer & Manufacturing Systems*
- Yu, Wang & Lai/ *FOREIGN-EXCHANGE-RATE FORECASTING WITH ARTIFICIAL NEURAL NETWORKS*
- Beyer et al/ *MARKOVIAN DEMAND INVENTORY MODELS*
- Shi & Olafsson/ *NESTED PARTITIONS OPTIMIZATION: Methodology And Applications*
- Samaniego/ *SYSTEM SIGNATURES AND THEIR APPLICATIONS IN ENGINEERING RELIABILITY*
- Kleijnen/ *DESIGN AND ANALYSIS OF SIMULATION EXPERIMENTS*
- Førsund/ *HYDROPOWER ECONOMICS*
- Kogan & Tapiero/ *SUPPLY CHAIN GAMES: Operations Management and Risk Valuation*
- Vanderbei/ *LINEAR PROGRAMMING: Foundations & Extensions, 3rd Edition*
- Chhajed & Lowe/ *BUILDING INTUITION: Insights from Basic Operations Mgmt. Models and Principles*
- Luenberger & Ye/ *LINEAR AND NONLINEAR PROGRAMMING, 3rd Edition*
- Drew et al/ *COMPUTATIONAL PROBABILITY: Algorithms and Applications in the Mathematical Sciences*

*** A list of the early publications in the series is at the end of the book ***

**FEASIBILITY AND INFEASIBILITY
IN OPTIMIZATION:
Algorithms and Computational Methods**

John W. Chinneck
Systems and Computer Engineering
Carleton University
Ottawa, Canada



John W. Chinneck
Carleton University
Ottawa, ON, Canada

Series Editor:
Fred Hillier
Stanford University
Stanford, CA, USA

Library of Congress Control Number: 2007935595

ISBN-13: 978-0-387-74931-0 (HB) ISBN-13: 978-0-387-74932-7 (e-book)

Printed on acid-free paper.

© 2008 by Springer Science+Business Media, LLC

All rights reserved. This work may not be translated or copied in whole or in part without the written permission of the publisher (Springer Science+Business Media, LLC, 233 Spring Street, New York, NY 10013, USA), except for brief excerpts in connection with reviews or scholarly analysis. Use in connection with any form of information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed is forbidden.

The use in this publication of trade names, trademarks, service marks and similar terms, even if they are not identified as such, is not to be taken as an expression of opinion as to whether or not they are subject to proprietary rights.

9 8 7 6 5 4 3 2 1

springer.com

For my parents, Mac and Shirley Chinneck, who fed that spark of curiosity with science books, telescopes and chemistry sets, even while I was busily dismantling useful household items, and for Linda and Annie, who make it all worthwhile.

Preface

Most applied optimization problems involve constraints: What is the maximum profit that a manufacturer can make given a limited number of machines and a limited labour force? What is the minimum amount of fuel that a fleet of trucks can consume while making a specified set of deliveries? What is the smallest amount of silicon needed to etch an electronic circuit while respecting limits on signal propagation time, inter-wire distance, etc.? Applications of constrained optimization are everywhere in industry, business, and government.

Of course, the solution returned by an optimization algorithm must also be feasible: we want the best possible value of the objective function that satisfies all constraints and variable bounds. Some optimization algorithms are not even able to proceed towards optimality until a feasible solution is available. In addition, the optimization question can be converted to a feasibility question, and vice versa. And what happens when an algorithm is unable to find a feasible solution? How do we know what went wrong? How do we repair the model? Questions of optimization, feasibility, and infeasibility are inextricably linked.

There has been a surge of important developments related to feasibility and infeasibility in optimization in the last two decades, a trend that continues to accelerate even today. New and more efficient methods for seeking feasibility in difficult optimization forms such as mixed-integer programs and nonlinear programs are emerging. The first effective algorithms for analyzing infeasible models have been discovered and implemented in commercial software. A community of researchers in constraint programming has begun to integrate their knowledge and approaches with the optimization community. Unanticipated spin-off applications of the new algorithms are being found. It's an exciting time.

The goal of this book is to summarize the state of the art in recent work at the interface of optimization and feasibility. It should serve as a useful reference for researchers, graduate students, and software developers working on optimization, feasibility, infeasibility, and related topics. Readers having a reasonable grounding in optimization (linear and nonlinear programming, mixed-integer programming, etc.) should have no difficulty following the material.

Lightweight coverage of topics in constraint programming, with an emphasis on constraint satisfaction problems, is included to illustrate the extensive overlap and convergence in the two literatures. An ideal version of the book would cover topics in constraint programming in the same depth as topics in optimization, but this is beyond the scope of this project: collecting and organizing the wealth of new developments relating to feasibility and infeasibility in optimization. I hope the resulting book is useful to both optimizers and constraint programmers, and

that it helps accelerate the ongoing merger of the two communities merge into a stronger hybrid.

Acknowledgements

My graduate work was conducted during the late 1970s and early 1980s. Inspired by the energy crises of those times, I constructed network optimization models to minimize the use of energy in large industrial plants. Later I found the optimization modeling more interesting than the energy aspects of this work. I had noticed that some of the processing network models that I was using in the energy work suffered from an inability to carry flow in some of the arcs, a pathology later labeled *nonviability* (see Sec. 9.2). I developed algorithms to automatically identify and analyze this problem.

Enter Harvey Greenberg. At that time he was involved in a project to develop an *Intelligent Mathematical Programming System* (IMPS) (see e.g. Greenberg (1996b)), and consequently had an interest in algorithms for analyzing modeling errors of various types, such as nonviability. Harvey organized an extraordinary series of meetings on the IMPS topic for an eclectic group of researchers from academia and industry. Harvey invited me to one of these meetings and, as they say, the rest is history. Sitting in the bar one night after the IMPS meeting we had a discussion about whether or not you could isolate the cause of infeasibility in a linear program to an irreducible subset of the constraints defining the model. At the time, Harvey didn't think it could be done, but I did, so I bet him a beer that I could find a way to do so. As you will see in Part II of the book, I won that bet.

But there is a postscript to this story. I have now known Harvey for around twenty years, and we have gone on to make numerous one-beer bets on other issues in optimization. I have not won a single one of those subsequent bets, so I am currently several hundred beers in debt to him. But I have an even bigger debt than that. Harvey became my unofficial mentor, always ready to provide advice and suggestions and listen to my ideas. His influence on my work has been profound.

Harvey and Pascal Van Hentenryck both took the time to read an early draft of the book and provide advice and suggestions that greatly improved it. Both pointed out topics that should be greatly expanded upon, especially the material on constraint programming, but time is unfortunately limited, so a full treatment of that topic remains another project. And as clever as those two fellows are, I'm sure I've managed to hide a few errors in the manuscript that they did not find: those are mine alone.

Last but not least are the two incredible ladies in my life, my wife Linda and daughter Annie, who can finally look at this book and see what kept me glued to the computer for such long hours over the past year. Thanks for being there.

John W. Chinneck

Table of Contents

Dedication.....	v
Preface.....	vii
Table of Contents.....	ix
List of Figures.....	xiii
List of Tables.....	xv
List of Algorithms.....	xvii
Introduction.....	xix
1. Preliminaries.....	1
1.1. The Optimization Model.....	1
1.2. Measuring Infeasibility.....	2
Part I: Seeking Feasibility.....	7
Model Reformulation.....	8
2. Seeking Feasibility in Linear Programs.....	11
2.1. The Phase 1 Algorithm.....	11
2.2. The Big- M Method.....	13
2.3. Phase 1 from Any Basis.....	13
2.4. Crash Start Heuristics.....	15
2.5. Crossover from an Infeasible Basis.....	16
2.6. Advanced Starts: Hot and Warm Starts.....	17
2.7. Seeking Feasibility and Optimality Simultaneously.....	18
2.8. Projection Methods.....	19
3. Seeking Feasibility in Mixed-Integer Linear Programs.....	23
3.1. Pivot-and-Complement and Pivot-and-Shift Heuristics.....	25
3.2. The OCTANE Heuristic.....	28
3.3. The Feasibility Pump.....	30
3.3.1. The Feasibility Pump for Mixed-Integer Nonlinear Programs.....	34
3.4. Branching Variable Selection by Active Constraints Methods.....	37
3.5. Conflict Analysis.....	42
3.6. Market Split Problems.....	43
4. A Brief Tour of Constraint Programming.....	45
4.1. Branching in the Satisfiability Problem.....	49

5.	Seeking Feasibility in Nonlinear Programs.....	51
5.1.	Penalty Methods.....	52
5.2.	Determining the Characteristics of an NLP.....	54
5.2.1.	Convex Sampling Enclosures.....	58
5.2.2.	Hit-and-Run Methods.....	59
5.2.3.	Approximating Nonconvex Feasible Regions.....	60
5.3.	Bootstrapping in a Convex Constrained Region.....	60
5.4.	Initial-Point Placement Heuristics.....	63
5.5.	Constraint Consensus Methods for Approximate Feasibility.....	65
5.6.	Finding a Good Sampling Box for Multistart.....	73
5.6.1.	Tightening the Variable Bounds.....	73
5.6.2.	Best Heuristic Sampling Box.....	76
5.7.	Multistart Methods.....	77
5.7.1.	MSNLP Feasibility Mode.....	80
5.7.2.	Multistart Constraint Consensus.....	80
5.8.	Bootstrapping Method of Debrosse and Westerberg.....	85
5.9.	Global Optimization.....	87
	Part II: Analyzing Infeasibility.....	89
6.	Isolating Infeasibility.....	93
6.1.	General Logical Methods.....	94
6.1.1.	Logical Reduction of Models and Presolving.....	95
6.1.2.	The Deletion Filter.....	97
6.1.3.	The Additive Method.....	98
6.1.4.	The Elastic Filter.....	101
6.1.5.	Speed-ups: Treating Constraints in Groups.....	104
6.1.6.	Speed-ups: Combining the Additive Method and the Deletion Filter.....	109
6.1.7.	Sampling Methods.....	110
6.2.	Methods Specific to Linear Programs.....	112
6.2.1.	The Reciprocal Filter.....	113
6.2.2.	The Sensitivity Filter.....	114
6.2.3.	Pivoting Methods.....	116
6.2.4.	Interior Point Methods.....	118
6.2.5.	Speed-ups: Combining Methods.....	118
6.2.6.	Guiding the Isolation.....	120
6.2.7.	Finding Useful Isolations.....	122
6.2.8.	Analyzing Infeasible Network LPs.....	127
6.2.9.	Software.....	128
6.3.	Methods Specific to Mixed-Integer Linear Programming.....	130
6.3.1.	A Deletion Filter for MIPs.....	133
6.3.2.	Additive Methods for MIPs.....	134
6.3.3.	An Additive/Deletion Method for MIPs.....	137
6.3.4.	Using the Information in the Initial Branch and Bound Tree.....	138

6.3.5.	Speed-ups.....	140
6.3.6.	Conclusions from Empirical Studies.....	140
6.3.7.	Software Survey.....	143
6.4.	Methods Specific to Nonlinear Programming.....	143
6.4.1.	Deletion Filtering.....	144
6.4.1.1.	Speeding the Isolation by Grouping Constraints	149
6.4.2.	IIS Isolation by the method of Debrosse and Westerberg.....	150
6.4.3.	Methods for Quadratic Programs.....	151
6.4.4.	Methods for Space-Covering Global Optimizers.....	153
6.4.5.	Software Survey.....	154
6.5.	Methods Specific to Constraint Programming.....	154
7.	Finding the Maximum Feasible Subset of Linear Constraints.....	159
7.1.	Exact Solutions.....	161
7.1.1.	An Exact Solution via MIP.....	161
7.1.2.	An Exact Formulation via Equilibrium Constraints	162
7.2.	IIS Enumeration and Covering.....	164
7.3.	Phase One Heuristics.....	167
7.4.	Chinneck's SINF-Reduction Heuristics	169
7.5.	Two-Phase Relaxation-Based Heuristic	179
7.6.	Randomized Thermal Relaxation Algorithms.....	181
7.7.	An Interior-Point Heuristic.....	183
7.8.	Working with IIS Covers.....	184
7.8.1.	Single Member IIS Covers.....	185
7.8.2.	Finding Specific IISs Based on IIS Covers	186
7.9.	The Minimum Number of Feasible Partitions Problem	189
7.10.	Partial Constraint Satisfaction in Constraint Programming.....	193
8.	Altering Constraints to Achieve Feasibility.....	197
8.1.	Shifting Constraints.....	197
8.1.1.	Using the Phase 1 Result.....	198
8.1.2.	Minimizing the l_1 Norm.....	199
8.1.3.	Least-Squares Methods.....	199
8.1.4.	Roodman's Bounds on Minimum Constraint Adjustments.....	200
8.1.5.	A Fuzzy Approach to Constraint Shifting	202
8.1.6.	A Goal Programming Approach to Constraint Shifting.....	202
8.1.7.	Constraint Shifting in Sequential Quadratic Programming.....	204
8.1.8.	Violating a Limited Number of Constraints by a Limited Amount.....	205
8.2.	Adjusting the Constraint Matrix.....	206
8.3.	Related Research.....	208

Part III: Applications	211
9. Other Model Analyses.....	213
9.1. Analyzing Unbounded Linear Programs.....	213
9.2. Analyzing the Viability of Network Models.....	213
9.3. Analyzing Multiple-Objective Linear Programs.....	216
9.3.1. Interaction Analysis of the Constraints.....	218
9.3.2. Interaction Analysis of the Objectives.....	218
9.3.2.1. Generating Different Interacting Sets of Objectives.....	220
9.3.2.2. Which Objectives Conflict With a Particular Objective?.....	221
9.3.2.3. Evaluating the Relative Amount of Objective Interference.....	221
9.3.3. Summary of the Method.....	222
9.3.4. Example.....	224
10. Data Analysis.....	227
10.1. Classification and Neural Networks.....	227
10.2. Data Depth.....	231
10.3. Errors in Massive Data Sets.....	232
11. Miscellaneous Applications.....	235
11.1. Radiation Treatment Planning.....	235
11.2. Protein Folding.....	236
11.3. Digital Video Broadcasting.....	237
11.4. Automated Test Assembly.....	238
11.5. Buffer Overrun Detection.....	239
11.6. Customized Page Ranking.....	239
11.7. Backtracking in Tree-Structured Search.....	240
11.8. Piecewise Linear Model Estimation.....	242
11.9. Finding Sparse Solutions to Systems of Linear Equations.....	243
11.10. Various NP-Hard Problems.....	244
12. Epilogue.....	247
References.....	249
Index.....	265

List of Figures

- Fig. P1.1. Typical easy-hard-easy pattern for determination of feasibility status ... 7
- Fig. 2.1. Several steps in a cyclic orthogonal projection method 21
- Fig. 2.2. The consensus vector (solid) results from the component-wise averaging of the three feasibility vectors (dashed) 21
- Fig. 3.1. Two-dimensional octagon around unit cube 29
- Fig. 3.2. Simplex iterations performance profiles with Cplex internal heuristics off (Patel and Chinneck 2006) 41
- Fig. 3.3. Simplex iterations performance profiles with all Cplex 9.0 heuristics turned on (Patel and Chinneck 2006) 42
- Fig. 5.1. Function shape is assessed via difference measurements along the line segment (Chinneck 2002) 56
- Fig. 5.2. Examples of convex and nonconvex region effects (Chinneck 2002) .. 57
- Fig. 5.3. Hit-and-run sampling in a convex enclosure (Chinneck 2002) 59
- Fig. 5.4. Example iteration of the Constraint Consensus method (Chinneck 2004) 66
- Fig. 5.5. Nonlinear interval analysis via sampling tightens the bounds on variable y . (a) inequality constraint. (b) equality constraint (Chinneck 2002) ..74
- Fig. 5.6. Range cutting (Chinneck 2002) 75
- Fig. 5.7. Constraint Consensus invocations on 2D Branin function (MacLeod and Chinneck 2007) 82
- Fig. 5.8. Bin voting 82
- Fig. 5.9. Theorem 5.1 85
- Fig. 5.10. Theorem 5.3 86
- Fig. 6.1. Simple linear IIS 93
- Fig. 6.2. Identifying an IIS by sampling 110
- Fig. 6.3. Sampling indicates infeasibility relative to constraint 111
- Fig. 6.4. The sensitivity filter 115
- Fig. 6.5. The branch and bound solution fails to terminate (Guieu and Chinneck 1999) 131
- Fig. 6.6. An infeasible MIP with feasible LP relaxation (Guieu and Chinneck 1999) 132
- Fig. 6.7. Constraints in NLTEST1 148
- Fig. 6.8. Sum of the absolute constraint violations in NLTEST1 148
- Fig. 6.9. Sum of absolute constraint violations for NLTEST1 MIS1 149
- Fig. 6.10. Example of method of Debrosse and Westerberg 150

- Fig. 7.1. Example infeasibility 169
- Fig. 7.2. A pathological counter-example (Chinneck 1996c) 172
- Fig. 9.1. Example processing node with ratio equations 214
- Fig. 9.2. A nonviable pure network 214
- Fig. 9.3. Finding a minimal nonviability in a processing network
model (Chinneck 1996b) 215
- Fig. 9.4. Examples of nonconflicting and conflicting objectives 219
- Fig. 10.1. Separating hyperplane 228
- Fig. 10.2. The grey point has data depth 2 231
- Fig. 11.1. A piecewise linear model consisting of three slabs 242

List of Tables

- Table 6.1. Characteristics of the netlib infeasible LPs 113
- Table 6.2. Summary of results for IS isolation methods for MIPs 141
- Table 6.3. Example IIS isolation by method of Debrosse and Westerberg (1973) 151
- Table 7.1. IIS cover cardinality on difficult LPs for two Phase 1 methods (Chinneck 1996c) 169
- Table 7.2. Comparison of algorithms on difficult infeasible LPs (Chinneck 2001a) 177
- Table 9.1. Normalized objective interference table 225
- Table 10.1. Classification data sets 228
- Table 10.2. Three algorithms for classification (Chinneck 2001a) 229
- Table 10.3. More MAX FS algorithms for classification (Amaldi et al. 2007) 230

List of Algorithms

- Alg. 3.1. General steps in the branch and bound method for solving MIPs 24
- Alg. 3.2. The pivot-and-complement heuristic for binary programs
(Balas and Martin 1980) 27
- Alg. 3.3. The pivot-and-shift integer-feasibility seeking search phase
(Balas et al. 2004)28
- Alg. 3.4. Main steps in the OCTANE heuristic (Balas et al. 2001) 29
- Alg. 3.5. Simplified feasibility pump algorithm 31
- Alg. 3.6. The feasibility pump for binary MIPs (Fischetti et al. 2005) 32
- Alg. 3.7. The feasibility pump for general MIPs (Bertacco et al. 2005) 33
- Alg. 3.8. The feasibility pump for convex mixed-integer nonlinear programs .. 35
- Alg. 5.1. Bootstrapping procedure to achieve initial feasibility of
a convex constrained region (Chinneck 2002) 62
- Alg. 5.2. The basic Constraint Consensus algorithm (Chinneck 2004) 67
- Alg. 5.3. Feasibility-distance based Constraint Consensus
(FDnear, FDfar) (Ibrahim and Chinneck 2005) 68
- Alg. 5.4. Average direction-based (DBavg) Constraint Consensus
(Ibrahim and Chinneck 2005) 69
- Alg. 5.5. Maximum direction-based (DBmax) Constraint Consensus
(Ibrahim and Chinneck 2005) 70
- Alg. 5.6. Direction-based and bound-based (DBbnd) Constraint
Consensus (Ibrahim and Chinneck 2005) 72
- Alg. 5.7. Bootstrapping method by Debrosse and Westerberg (1973) 88
- Alg. 6.1. The deletion filter 97
- Alg. 6.2. The additive method 98
- Alg. 6.3. The dynamic reordering additive method 100
- Alg. 6.4. The elastic filter 102
- Alg. 6.5. The depth first binary search filter 105
- Alg. 6.6. The generalized binary search filter 107
- Alg. 6.7. The additive/deletion method 109
- Alg. 6.8. The sensitivity filter 114
- Alg. 6.9. The deletion/sensitivity filter 119
- Alg. 6.10. The additive/sensitivity method 120
- Alg. 6.11. The (*IR-LC-BD*) deletion filter for MIPs 133

• Alg. 6.12. The basic additive method for MIPs	135
• Alg. 6.13. Dynamic reordering additive method for MIPs	136
• Alg. 6.14. Basic additive/deletion method for MIPs	137
• Alg. 6.15. Using information from the original branch and bound tree	139
• Alg. 6.16. The deletion filter for NLPs	146
• Alg. 6.17. Using the MIS	147
• Alg. 6.18. Finding the set of killing constraints	152
• Alg. 6.19. Deletion filtering for inequality-constrained QCQP	153
• Alg. 7.1. Minimum-weight IIS set covering algorithm (Parker and Ryan 1996)	164
• Alg. 7.2. The constraint frequency heuristic for the IIS cover (Tamiz et al. 1995)	166
• Alg. 7.3. Heuristic 1 for MIN IIS COVER (Chinneck 1996c)	171
• Alg. 7.4. Heuristic 2 for MIN IIS COVER (Chinneck 2001a)	176
• Alg. 7.5. Overall logic of the two-phase relaxation-based heuristic (Amaldi et al. 2007)	180
• Alg. 7.6. The IIS member labelling scheme	185
• Alg. 7.7. Finding one IIS for each member of the IIS set cover	187
• Alg. 7.8. Finding a single IIS given the complete set of IIS covers (Liffiton and Sakallah 2005)	188
• Alg. 7.9. Backtracking greedy algorithm for MIN PFS (Bemporad et al. 2005)	192
• Alg. 7.10. Partition refinement (Bemporad et al. 2005)	194
• Alg. 8.1. Adjusting the α and β conditions (Censor et al. 2006)	206
• Alg. 9.1. Analyzing MOLPs using IIS isolation algorithms	223

Introduction

To be, or not to be: that is the question...
From *Hamlet* by William Shakespeare

Shakespeare certainly hit the nail on the head: the most basic question of all is whether or not something exists: an object, a person or a solution that satisfies a given set of constraints. For Shakespeare, human existence was a fundamental question of life; for this book, existence of a feasible solution is a fundamental question of optimization.

Why such interest in feasibility and infeasibility in the context of optimization? Surely it is most important to find the best (i.e. optimum) solution, rather than just any feasible solution? The questions of feasibility and optimality are in fact two sides of the same coin. First, the existence of a feasible solution is a very fundamental question: before you can determine which solution is the best, you must first determine whether or not it is even possible for a feasible solution to exist at all. Second, it is easy to convert an optimization question to a feasibility question (and vice-versa), so the two questions are fundamentally the same. For example, you can pose the feasibility question as to whether or not a solution exists with an objective function value that is at least as good as a certain stated aspiration value. Over a series of iterations this aspiration value can be adjusted until we can definitely answer that no solution exists beyond a certain value. That last feasible solution is the optimum value of the objective function, found by answering a series of feasibility questions.

Looking at this the opposite way, it is common practice to pose feasibility questions as optimization problems. This is the basic idea of any phase 1 technique: create an objective function that measures the degree of violation of the constraints at any given point, and then minimize this function. If a value of zero can be found for this phase 1 optimization problem, then a feasible point exists, otherwise the model may not be feasible.

Third, there are unique and interesting questions associated with feasibility and infeasibility in optimization. For example, given a set of constraints that a solver determines to be infeasible, provide a diagnosis of why this is so. This question has grown in importance in recent years as optimization models have grown larger and more complex in step with the phenomenal increases in inexpensive computing power. One approach to this question is to isolate an *irreducible infeasible subset (IIS)* of the constraints, i.e. a (small) subset of constraints that is itself infeasible, but becomes feasible if one or more constraints is removed. This helps focus the diagnosis and model repair efforts and is especially helpful in very

large models. This approach is well summarized by Greenberg's aphorism: "diagnosis = isolation + explanation" (Greenberg 1993). A related diagnostic question is this: given an infeasible model, what is the smallest number of constraints to remove such that the remaining constraints constitute a feasible set? Another is: what is the best way to repair the infeasible system (e.g. what is the smallest set of changes that can be made to the constraint right hand sides such that the set of constraints becomes feasible)?

Many of the algorithms used in answering these diagnostic questions depend on assessing the feasibility of numerous subsets of the original set of constraints. Hence those algorithms operate much more efficiently if the feasibility status of an arbitrary set of constraints can be determined quickly (which is of course a fundamental feasibility question itself). This is not difficult for sets of linear constraints, but it can be extremely difficult and time-consuming to determine feasibility status at all when there are nonlinear constraints or integer restrictions on some or all of the variables. Hence one focus of this book is algorithms for improving the speed with which the first feasible solution can be found (if one exists) for the more difficult cases in optimization.

A fourth major reason for interest in feasibility-related algorithms is the many applications that have been found for them. Some of these applications are surprising: data classification, training of neural networks, radiation treatment planning, analysis of protein folding, automatic test assembly, applications in statistics, etc. Some of these are briefly reviewed in Part III.

Finally, the question of feasibility or infeasibility is a major overlap between the field of optimization and the field of constraint programming. Constraint programming, arising from computer science, has special strength in seeking a yes/no answer to the question of whether a solution exists for a stated set of constraints; this is identical to the feasibility question in optimization. However, because of their different roots and traditions, constraint programming researchers approach the question in a different way and with different techniques. The two fields have begun to merge in recent years, resulting in stronger hybrid techniques. Constraint programming techniques and their links with optimization are addressed at an elementary level.

The emphasis in this volume is on algorithms and computational methods, specifically practical algorithms for solving the feasibility/infeasibility related problems that are the main subject. The book summarizes the main developments over the last twenty years or so, a very active period for the field, spurred by improvements in computing power and an increase in the size and complexity of optimization models. It should prove useful for academics teaching and conducting research in the field and their graduate students, as well as practitioners.

As opposed to a mathematical treatment, we take the involvement of a computer as a given: modern optimization problems are normally of such scale and complexity that they simply cannot be solved without using a computer. The essential element in solving a feasibility or optimization problem via computer is an efficient and effective algorithm. The computer implementation of these algorithms introduces a number of practical issues and complications, such as tolerances. These are also dealt with as they arise.

A Note on Theorems: There is a significant amount of mathematical development underlying the algorithms and computational methods that are the main topic of this book. To keep the focus on algorithms, proofs are generally included where a theorem relates to whether an algorithm functions as intended. However, where theorems relate to mathematical underpinnings, the proof is generally omitted in favour of a simple reference to the original publication containing the proof.