Feasibility and Infeasibility in Optimization: Algorithms and Computational Methods

John W. Chinneck

Springer International Series in Operations Research & Management Science, Vol. 118.
Approximately 300 pages, hardcover. ISBN: 978-0-387-74931-0. Available December 1, 2007.

Publishers web page:
http://www.springer.com/west/home/new+%26+forthcoming+titles+%28default%29?SGWID=4-40356-22-173757940-0

About this book

Constrained optimization models are core tools in business, science, government, and the military with applications including airline scheduling, control of petroleum refining operations, investment decisions, and many others. Constrained optimization models have grown immensely in scale and complexity in recent years as inexpensive computing power has become widely available. Models now frequently have many complicated interacting constraints, giving rise to a host of issues related to feasibility and infeasibility. For example, it is sometimes difficult to find any feasible point at all for a large model, or even to accurately determine if one exists, e.g. for nonlinear models. If the model is feasible, how quickly can a solution be found? If the model is infeasible, how can the cause be isolated and diagnosed? Can a repair to restore feasibility be carried out automatically? Researchers have developed numerous algorithms and computational methods in recent years to address such issues, with a number of surprising spin-off applications in fields such as artificial intelligence and computational biology. Over the same time period, related approaches and techniques relating to feasibility and infeasibility of constrained problems have arisen in the constraint programming community.

Feasibility and Infeasibility in Optimization is a timely expository book that summarizes the state of the art in both classical and recent algorithms related to feasibility and infeasibility in optimization, with a focus on practical methods. All model forms are covered, including linear, nonlinear, and mixed-integer programs. Connections to related work in constraint programming are shown. Part I of the book addresses algorithms for seeking feasibility quickly, including new methods for the difficult cases of nonlinear and mixed-integer programs. Part II provides algorithms for analyzing infeasibility by isolating minimal infeasible (or maximum feasible) subsets of constraints, or by finding the best repair for the infeasibility. Infeasibility analysis algorithms have arisen primarily over the last two decades, and the book covers these in depth and detail. Part III describes applications in numerous areas outside of direct infeasibility analysis such as finding decision trees for data classification, analyzing protein folding, radiation treatment planning, automated test assembly, etc.

A main goal of the book is to impart an understanding of the methods so that practitioners can make immediate use of existing algorithms and software, and so that researchers can extend the state of the art and find new applications. The book is of interest to researchers, students, and practitioners across the applied sciences who are working on optimization problems.

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 (dased) 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