

UC Berkeley

UC Berkeley Electronic Theses and Dissertations

Title

Branch-and-Cut for Nonlinear Power Systems Problems

Permalink

<https://escholarship.org/uc/item/7nh5p9ff>

Author

Chen, Chen

Publication Date

2015

Peer reviewed|Thesis/dissertation

Branch-and-Cut for Nonlinear Power Systems Problems

by

Chen Chen

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Engineering / Industrial Engineering and Operations Research

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Shmuel S. Oren, Chair

Professor Alper Atamtürk

Professor Kameshwar Poolla

Fall 2015

Branch-and-Cut for Nonlinear Power Systems Problems

Copyright 2015
by
Chen Chen

Abstract

Branch-and-Cut for Nonlinear Power Systems Problems

by

Chen Chen

Doctor of Philosophy in Engineering / Industrial Engineering and Operations
Research

University of California, Berkeley

Professor Shmuel S. Oren, Chair

This dissertation is concerned with the design of branch-and-cut algorithms for a variety of nonconvex nonlinear problems pertaining to power systems operations and planning. By understanding the structure of specific problems, we can leverage powerful commercial optimization solvers designed for convex optimization and mixed-integer programs. The bulk of the work concerns the Alternating Current Optimal Power Flow (ACOPF) problem. The ACOPF problem is to find a minimum cost generation dispatch that will yield flows that can satisfy demand as well as various engineering constraints. A standard formulation can be posed as a nonconvex Quadratically Constrained Quadratic Program with complex variables. We develop a novel spatial branch-and-bound algorithm for generic nonconvex QCQP with bounded complex variables that relies on a semidefinite programming (SDP) relaxation strengthened with linear valid inequalities. ACOPF-specific domain reduction or bound tightening techniques are also introduced to improve the algorithm's convergence rate. We also introduce second-order conic valid inequalities so that any SDP can be outer-approximated with conic quadratic cuts and test the technique on ACOPF. Another application is the incorporation of convex quadratic costs in unit commitment, which is a multi-period electric generation scheduling problem. We show that conic reformulation can both theoretically and practically improve performance on this mixed-integer nonlinear problem. We conclude with methods for approximating a mixed-integer convex exponential constraint. Applications include capital budgeting, the system reliability redundancy problem, and feature subset selection for logistic regression.

To my family. It is thanks to their support and encouragement that I have had the privilege to spend the last five years pursuing my curiosity.

Contents

Contents	ii
List of Figures	iv
List of Tables	v
1 Overview	1
1.1 Optimization	1
1.2 Convex Optimization	2
1.3 Branch-and-Bound	4
1.4 Relaxations and Cuts	7
1.5 Chapter Summaries	9
2 Spatial Branch-and-Cut for Complex Bounded QCQP	11
2.1 Introduction	11
2.2 Spatial Branch-and-Cut	13
2.3 Computational Experiments	29
2.4 Conclusion	37
3 Bound Tightening for the Alternating Current Optimal Power Flow Problem	38
3.1 Introduction	38
3.2 Formulations	40
3.3 New Instances With Large Duality Gap	41
3.4 Bound Tightening Procedures for ACOPF	43
3.5 Computational Experiments	52
3.6 Conclusion	53
4 Sparse Cuts for a Positive Semidefinite Constraint	55
4.1 Linear Valid Inequalities	55

4.2	Second-Order Cone Valid Inequalities	56
4.3	Positive Semidefinite Completion	63
4.4	Computational Experiments	66
4.5	Conclusion	71
5	Unit Commitment with Quadratic Costs	72
5.1	Introduction	72
5.2	Formulation	73
5.3	Quadratic Costs and Conic Reformulation	75
5.4	Computational Experiments	76
5.5	Conclusion	78
6	On Mixed-Integer Geometric Programming	79
6.1	Introduction	79
6.2	Outer-Approximation of the Exponential Constraint	80
6.3	Submodular Inequalities	84
6.4	Application: Capital Budgeting	86
6.5	Application: System Reliability Redundancy	91
6.6	Application: Feature Subset Selection for Logistic Regression	95
6.7	Conclusion	97
7	Conclusion	98
7.1	Summary	98
7.2	Extensions: Applications	99
7.3	Extensions: Methodology	99
A	Appendix	102
A.1	Creating Bounds for CSDP from CQCQP	102
	Bibliography	104

List of Figures

2.1	Clockwise, starting from top-left: \mathcal{J}_R ; cone of (2.12a), upper bound of (2.12b), and lower bound of (2.12c); cone of (2.12a) and valid inequality (2.6a); cone of (2.12a) and valid inequality (2.6b).	25
6.1	$y * (1 - \log(y))$	89

List of Tables

2.1	Comparison of branching rules using CSDP for ACOPF	33
2.2	Comparison of branching rules using CSDP+VI for ACOPF	33
2.3	Comparison of relaxations on BoxQP instances	34
2.4	Comparison of branching rules with SDP+RLT relaxation on BoxQP . . .	35
2.5	Comparison of branching rules with SDP+VI relaxation on BoxQP	36
3.1	Comparison with and without bound tightening	53
3.2	Breakdown of time spent (seconds)	54
3.3	Comparison of branching rules using CSDP+VI and bound tightening for ACOPF	54
4.1	Solution times (s)	68
4.2	Relaxation Optimum as Percent of Best Primal Optimum	68
4.3	Clique Size Distribution	68
4.4	Capturing the Strength of 2x2 Complex PSD Inequalities	70
4.5	Adding PSD Cuts to the 2×2 Relaxation	70
5.1	Unit Commitment with Varying Quadratic Cost Coefficients	77
6.1	Exponential Approximation Error (percent)	83
6.2	Replicated Capital Budgeting Instances: B&B data	87
6.3	Replicated Capital Budgeting Instances: Gaps	88
6.4	Gradient cuts on small instances	91
6.5	$\alpha = 1, \beta = 0.5, m = 25, n = 25$	91
6.6	Reliability Instances	94

Acknowledgments

I have had the honour of working with two tremendous advisors: Professors Alper Atamtürk and Shmuel Oren. Professor Oren, being part of your lab has felt like being part of a family. The impact of your work in the energy industry is inspiring, and your opinions have been invaluable throughout my research. Professor Atamtürk, thanks to your mentorship I've learned to develop a critical eye that will help me throughout my life. Your input has always been spot-on, and I hope one day to attain your level of discernment. I would also like to thank the other members of my qualifying and dissertation committee: Professors David Tse, Ilan Adler, and Kameshwar Poolla.

My family's support has played a major part in helping me complete this dissertation. My love and gratitude to my parents, Xiu Hong and Zhi Qing, is beyond words. My brother Tao and his wife Linh have always been there for me. In turn, I will look out for the newest additions to our family: my niece Julia and nephew Reynard.

One of the fascinating things about academia is the great diversity in approaches to research. I have tried my best to learn from the masters; I would like to mention a few here. Classes and discussions with Professor Ilan Adler showed me the tremendous power and elegance of understanding and organizing topics using a handful of core mathematical principles. Much of my work so far has been groping in the dark, but I aim to develop this kind of theoretical maturity in time. Similarly, Professor Papadimitriou's class on Algorithmic Game Theory was a profound experience. His level of intuition allows him to have a poetic understanding of complexity, and it opened my eyes to see how theory in one field could meaningfully impact another. Professors Andrew Lim and Max Shen were always eager to discuss ideas, and I am grateful for the inspiration that they provided, particularly in my first year. Professors outside Berkeley who have provided helpful guidance and mentorship include Daniel Bienstock at Columbia and professors Roy Kwon and Timothy Chan from my alma mater, the University of Toronto.

Perhaps the greatest benefit of studying at Berkeley is day-to-day interaction with fellow graduate students. I have learned so much from my fellow students, and virtually every idea herein has been sanity-checked by at least one of you. From Alper's lab: Andreas Gomez, Avinash Bhardwaj, Birce Tezel, Carlos Deck, Qi Zhang, and Peng Yi. From Shmuel's lab: Anthony Papavasiliou, Clay Campaigne, Kory Hedman, Jiaying Shi, Paula Lipka, Ruoyang Li, Tanachai Limpaitoon, and Zhu Yang. I thank you for being there, for the many shared trials and tribulations, long days and late nights. Thanks to some other IEOR graduate students for the good

times and compelling discussions: Amber Richter, Animesh Garg, Auyon Siddiq, Evan Davidson, Rebecca Sarto, Stewart Liu, Tugce Gurek, and Yong Liang.

In the summer of my first year I went to the Federal Energy Regulatory Commission to work with the chief economic adviser, Dr. Richard O'Neill. With his support and guidance I developed the initial seeds of many of the ideas in this thesis. He is a tremendous person and I have learned a great deal from him and from the rest of his team: Anya Castillo, Brian Marrs, Eric Krall, Mary Cain, Thomas Dautel.

I would like to thank my friends for being there for me, both personally, and for bouncing off research ideas. My roommates: Michael Kim, Shelly Manber, Clement Camp, and Yonatan Mintz. My friends from the economics department: Casper Jørgensen, David Birke, Eric Auerbach, Jeffrey Kang, Jordan Ou, Kaushik Krishnan, Marion Aouad, Mikkel Sølvsten, Sandile Hlatshwayo, Vincenzo Pezone, Waldo Tekempa. Among those friends but not in the department are Lene Sønder and Sylvan Herskowitz. I lived in the International House in my first year, and have made lifelong friends there, including: Andrew Critch, Daniel Hogan, Jeff Schauer, Michael Pacer, and Mahendra Prasad. I also cannot forget my roots, my friends from Toronto, including: Igor Nikiforovski, Ines Fernandez, Jonathan Lu, Justin Shum, Marc Tyndel, Mark Jeffrey, Paolo del Puerto, Ronald Lo, Velibor Misic, and Xavier Tang.

Finally, this research would not be possible without funding from the following organizations: the PGS scholarship from NSERC, and research funding from the NSF, the Department of Defense, and FERC.

Chapter 1

Overview

This dissertation is concerned with the design of algorithms for nonlinear nonconvex optimization problems that arise in power systems planning and operations. The general philosophy is to adopt tried-and-true techniques for branch-and-cut in mixed-integer linear programming (MILP) and extend them to nonlinear problems. The intention is to develop practical methods by crafting tools around problem-specific structures. These structures come from various applications that are relevant to the power industry, such as electric generation scheduling and dispatch, and reliability planning. Power systems engineering is a rich field of study in its own right, but discussion throughout will be primarily centered around the design of algorithms. The remainder of this overview will provide some background on topics in optimization relevant to the dissertation, as well as a summary of the remaining chapters. Some familiarity with computational complexity theory, graph theory, and linear algebra is assumed of the reader. Applying complexity theory, which is classically built around decision problems, to optimization problems may require some modest additional background; the reader is referred to De Klerk [45] on this matter.

1.1 Optimization

We shall adopt the mathematical programming perspective of optimization:

$$\text{minimize } f(x) \tag{1.1a}$$

$$\text{subject to } g_j(x) \leq 0, j = 1, 2, \dots, m, \tag{1.1b}$$

$$x \in \Omega. \tag{1.1c}$$

x is an n -dimensional decision vector of unknowns, and $f, g_j, j = 1, 2, \dots, m$ are real-valued functions of x . Ω is a subset of n -dimensional space, typically \mathbb{R}^n in the case of nonlinear programming (e.g. [106]), or $\mathbb{R}^k \times \mathbb{Z}^l, k + l = n$ in the case of mixed-integer programming (e.g. [115]). A solution x is said to be feasible if it satisfies the constraints (1.1b)-(1.1c), and a problem instance is feasible if there exists a feasible solution. Let $z^* = \inf\{f(x) | g_j(x) \leq 0 \forall j, x \in \Omega\}$, and suppose we are given an optimality tolerance parameter $\epsilon > 0$. An optimization problem is solved when one of three conditions can be proven: 1) an ϵ -optimal solution is found, which is a feasible solution x_ϵ^* such that $f(x_\epsilon^*) \leq z^* + \epsilon$; 2) the problem is shown to be unbounded, $z^* = -\infty$; 3) the problem is shown to be infeasible, $z^* = +\infty$.

The problem as posed cannot be solved in general. For instance, one can set $\Omega = \mathbb{Z}^n, f = 0, m = 2$ and let g_1 be a polynomial function with integer coefficients and let $g_2 = -g_1$. This is Hilbert's tenth problem, which is undecidable [44]. Even with restrictions to ensure solvability one can face difficulties with computational scalability. For instance, solving the pure integer linear programming problem, where f, g_j are linear with rational coefficients and $\Omega = \mathbb{Z}^n$, is strongly NP-complete [127]. Hence there is a tradeoff between expressivity and computational tractability. In practice, one may forego the stringent requirements of solving. One concession is to set high optimality tolerance parameter. In the case of nonconvex nonlinear constraints one may also need to sacrifice feasibility, and accept instead feasibility to within some numerical tolerance (e.g. [26]). Classical nonlinear programming is centered on analysis regarding convergence towards a local optimal, as characterized by the Karush-Kuhn-Tucker conditions, which is a weaker notion than global optimality (e.g. [106]). Hence an optimization algorithm may be described as *global* to distinguish it from a local or heuristic method that does not guarantee convergence to ϵ -optimality. Mixed-integer programming research is more concerned with global optimization of NP-hard problems, and so algorithms with poor theoretical worst-case convergence rates may be adopted provided they have practical performance. The next section will describe a broad class of problems that are computationally tractable for a given global optimality and feasibility tolerance.

1.2 Convex Optimization

Consider the convex optimization problem:

$$\min c^T x \tag{1.2a}$$

$$\text{s.t. } x \in C. \tag{1.2b}$$

We have a linear objective function (1.2a) with coefficients $c \in \mathbb{R}^n$, and C is a closed convex domain in \mathbb{R}^n . This problem is NP-hard in general (e.g. [48]), but if some mild assumptions about C hold the problem can be solved by algorithms that run in polynomial-time with respect to a fixed ϵ [32]. Among these algorithms are interior point methods (e.g. [117]), which have good practical performance. For certain types of convex optimization problems interior point methods can guarantee an ϵ -optimal solution in time polynomial in the bit-size of the problem data and $\log(1/\epsilon)$ [116]. A standard description of such problems would involve conic programming (e.g. [117]), as the primal-dual interior point method leverages conic duality. However for brevity we shall avoid this theoretical route and focus instead on modeling structure. Consider the conic geometric program, a type of convex optimization problem:

$$\text{(CGP)} : \min c^T x \quad (1.3a)$$

$$\text{s.t. } A_0 x + b_0 \geq 0, \quad (1.3b)$$

$$\|A_j x + b_j\|_2 \leq d_j^T x + h_j, \quad j = 1, \dots, m, \quad (1.3c)$$

$$Q_0^{(k)} + \sum_{i=1}^n x_i Q_i^{(k)} \succeq 0, \quad k = 1, \dots, p, \quad (1.3d)$$

$$\exp(u_\ell^T x - s_\ell) \leq r_\ell^T x + t_\ell \quad \ell = 1, \dots, q. \quad (1.3e)$$

Constraints (1.3b) are linear inequality constraints, with $A_0 \in \mathbb{R}^{m \times n}$, $b_0 \in \mathbb{R}^m$ and \geq representing an element-wise inequality. If only linear constraints used, then we have the linear programming (LP) problem. LP can be solved exactly ($\epsilon = 0$) in polynomial time [79, 76] in the bit-model of the problem.

Constraints (1.3c) are second-order cone or conic quadratic constraints, where the data are of appropriate dimensions, $A_j \in \mathbb{R}^{m \times n}$, $b_j, d_j \in \mathbb{R}^n$, $h_j \in \mathbb{R} \forall j$, and $\|\cdot\|_2$ denotes the Euclidean norm. If we restrict ourselves to constraints of type (1.3c), then we have a Second Order Cone Programming (SOCP) or Conic Quadratic Programming problem. Note that linear inequalities are a special case where A_j, b_j have all zero entries, so SOCP includes LP as a special case. Second-order cone constraints can be used to model convex quadratic constraints. A variety of nonconvex constraints (see [101, 7]) can be modeled with SOCP, often using transformation with the rotated second-order cone:

$$u^2 \leq v_1 v_2, \quad v_1, v_2 \geq 0 \iff \|(2u, v_1 - v_2)\| \leq v_1 + v_2.$$

Constraints (1.3d) are linear matrix inequalities. We have symmetric real data matrices $Q_i^{(k)} \in \mathbb{S}^n \forall i, k$, and a condition $\succeq 0$ that the weighted sum of Q matrices

be positive semidefinite. If only linear matrix inequalities are used for constraints, then we have a semidefinite programming problem (SDP) in inequality form. SDP can be used to model second-order cone constraints (e.g. [147]), so SDP generalizes both LP and SOCP. The equality form of SDP uses a matrix of decision variables:

$$\begin{aligned} \text{(SDP)} : \min & \langle C, X \rangle \\ \text{s.t.} & \langle Q^{(k)}, X \rangle = b_k, k = 1, \dots, p \\ & X \succeq 0 \end{aligned}$$

$\langle \cdot, \cdot \rangle$ denotes the trace operator. Any SDP in equality form can be transformed to inequality form and vice-versa (e.g. [163, p. 5]).

Constraints (1.3e) are a generalization of the canonical constraints of geometric programming in convex form, where the right-hand side is a constant. The convex exponential constraint can be used to model risk (e.g.[13]), economic utility (e.g. [5]), and a host of engineering problems after some transformations (e.g. [41, 33]).

Chandrasekaran and Shah [37] show that an interior point method can solve CGP in polynomial-time, provided there exists a strictly feasible point, that is $\exists x \in \text{int}(C)$. This is a standard constraint qualification for the nonlinear problems SOCP, SDP, and GP that ensures interior-point method convergence. Infeasibility can be detected by interior point methods in polynomial time provided the problem is strictly infeasible [118], i.e remains infeasible after a small perturbation in the data. In SDP, for instance, a *weakly* feasible problem is one for which the feasible set is empty, but for any $\epsilon > 0$ there exists $X_\epsilon \succeq 0$ such that $|\langle Q^{(k)}, X_\epsilon \rangle - b_k| \leq \epsilon \forall k$. The complexity of detecting weak-infeasibility in SDP is an open question (e.g. [84, 128]).

Note that CGP is not an exhaustive example, as there are other types of constraints amenable to interior point (e.g. [116]). Moreover, interior point is not the only polynomial-time algorithm available for these problems. For instance, similar convergence results for solving a strictly feasible or strictly infeasible problem apply to the ellipsoid method [56].

1.3 Branch-and-Bound

Branch-and-bound is an algorithmic framework for optimization that was first proposed by Land and Doig [90] for integer linear programming . The method relies on a branching rule, which for a given set S will split the region into subsets $S_1, S_2, \dots, S_n \subseteq S$ so that $\bigcup_{i=1}^n S_i = S$. Denote $S_1^{(1)}$ to be the set of feasible solutions for the optimization problem to be solved, and denote $S_1^{(1)}, S_2^{(1)}, \dots, S_n^{(1)}$ to be the

subsets produced by branching. Recursive application of branching on $S_1^{(1)}$ will produce a search tree, where each child node has a different subset of the parent node's feasible set, and the initial or root node has $S_1^{(1)}$. The superscript of S denotes the depth of the search tree, and the subscript denotes the order in which the node was generated with respect to its depth.

We also require bounding procedures that will provide an upper bound ($z_U^*(S)$) with associated solution $x_U^*(S)$ and lower bound ($z_L^*(S)$) on $z^*(S) = \inf\{f(x)|x \in S\}$. We can, for instance, trivially set $z_U^*(S) = z_L^*(S) = f(x)$, $x_U^*(S) = x$ if S is a singleton, and $z_U^*(S) = +\infty$, $x_U^*(S) = \emptyset$, $z_L^*(S) = -\infty$ otherwise. Suppose the branching rule forms a partition and $S_1^{(1)}$ is itself finite. Then, provided f is computable to arbitrary numerical precision, the problem can be solved in finite time by complete enumeration of the leaf nodes of a search tree of sufficient depth. This is a brute-force or exhaustive search. Denote z_{GUB}^* to be the least upper bound encountered at any node of the search tree. The refinement of branch-and-bound is that we may be able avoid certain branches. If at any node we have $z_L^*(S_j^{(i)}) + \epsilon > z_{GUB}^*$, then branching is no longer needed, and the node is said to be pruned and is added to the leaf node set \mathcal{F} . Thus a good lower and upper bound function will result in a smaller search tree, resulting in implicit rather than complete enumeration. Note that a global lower bound z_{GLB}^* on the optimal objective $z^*(S_1^{(1)})$ is obtained from $z_{GLB}^* = \max\{z_L^*(S)|S \in \mathcal{F}\}$. Any node that has not been pruned or branched on is in the set of live nodes, \mathcal{N} . Each iteration of branch-and-bound selects a live node to prune or branch on; common node selection or tree traversal rules are depth-first search and breadth-first search using $z_L^*(S)$ as the criterion. An outline of branch-and-bound follows:


```

1. GET  $S_1^{(1)}$ 
   SET  $z_{GUB}^* := z_U^*(S_1^{(1)}), x_{GUB}^* := x_U^*(S_1^{(1)}), z_{GLB}^* = -\infty, \mathcal{N} = S_1^{(1)}, \mathcal{F} = \emptyset$ 
2. DETERMINE with node selection rule a live node  $S_k^{(j)}$ 
3.  $\mathcal{N} := \mathcal{N}/S_k^{(j)}$ 
   COMPUTE bounds  $x_U^*(S_k^{(j)}), z_U^*(S_k^{(j)}), z_L^*(S_k^{(j)})$ 
   IF  $z_U^*(S_k^{(j)}) < z_{GUB}^*$ 
      $z_{GUB}^* := z_U^*(S_k^{(j)}), x_{GUB}^* := x_U^*(S_k^{(j)})$ 
   ENDIF
   IF  $z_L^*(S_k^{(j)}) + \epsilon \leq z_{GUB}^*$ 
     BRANCH:  $\mathcal{N} := \mathcal{N} \bigcup_{r=\ell}^m S_r^{(j+1)}$ 
   ELSE
     PRUNE:  $\mathcal{F} := \mathcal{F} \cup S_k^{(j)}$ 
   ENDIF
4. IF  $\mathcal{N} \neq \emptyset$ 
   GO TO 2
   ELSE
     SET  $z_{GLB}^* := \max\{z_L^*(S) | S \in \mathcal{F}\}$ 
     PRINT  $x_{GUB}^*, z_{GUB}^*, z_{GLB}^*$ 
   ENDIF

```

Practical termination criteria can also be added due to limitations of time and space. Typical rule include pruning all nodes past a certain depth, and pruning all nodes after a time or memory limit has been reached.

We have so far avoided specifics regarding how one should implement branching and bounding. There are many possible branching and node selection rules (e.g. [2, 19]), and the correct choice depends on the nature of the problem, especially the types of constraints used. However, the corresponding theoretical analysis of branching is relatively limited (e.g. [170, 93]), so it remains perhaps more of an art than a science. The literature on finding upper bounds is truly vast, as any algorithm that is intended to produce a feasible solution can be employed. In mixed-integer programming a standard approach is to use rounding heuristics (e.g. [1]).

However, other possibilities for branch-and-bound include metaheuristics (e.g. [82, 132]), problem-specific greedy algorithms (e.g. [38]), or iterative algorithms designed for nonlinear programming. Furthermore, there is also the question of when to calculate a new upper bound (at every node, every k nodes, etc.). Lower bound calculation in branch-and-cut is typically performed by solving relaxations, to be discussed in the next section.

Branch-and-bound can be combined with other approaches to form a hybrid algorithm. For instance, since branch-and-bound is a search procedure, then reduction of the search space (the root problem's feasible set) can speed up convergence. It is not necessary to set the root problem $S_1^{(1)}$ as the set of feasible solutions; it suffices to have a set include at least one ϵ -optimal solution (should one exist). This is known as preprocessing, where work is done on the root problem before expanding the search tree. A constraint programming [136] algorithm, for instance, may be able to solve the optimization problem on its own, but it can also be employed to partially reduce the domain during preprocessing and pass the remaining problem to branch-and-bound (e.g. [3]). Reduction of the problem domain can be applied at any node, and the resulting algorithm can be called branch-and-reduce [139]. One can also decompose the initial problem in terms of its variables, including variables as needed in a method called branch-and-price [18]. Constraints can also be introduced as needed along the search tree in a procedure called branch-and-cut (e.g. [125]).

1.4 Relaxations and Cuts

As discussed in Section 1.3, a good lower bound helps to speed up the search for an ϵ -optimal upper bound. It will simplify the presentation to consider a problem with linear objective function: $z_S^* := \inf\{c^T x \mid x \in S\}$, $S \subseteq \mathbb{R}^n$. For general optimization this is without loss of generality as we can always replace the objective function with a new variable x_{n+1} and the constraint $f(x) \leq x_{n+1}$. Let us define a *relaxation* of the problem as $z_R^* := \inf\{c^T x \mid x \in R\}$ when $R \supseteq S$. It is clear that a relaxation provides a lower bound, i.e. $z_R^* \leq z_S^*$. A relaxation can be strengthened with valid inequalities, which are constraints that are satisfied by every point in S . A *cut* is a constraint that removes at least one point from R , and is assumed to be valid unless stated otherwise.

The relaxation should be computationally tractable if it is to be incorporated into branch-and-cut. From Section 1.2 we know that a broad class of convex optimization problems can be solved in polynomial-time. The convex hull of a subset $S \subseteq \mathbb{R}^n$, denoted $\text{conv}(S)$ is the intersection of all convex sets that contains S . It is therefore the strongest possible convex relaxation. The closed convex hull $\text{clconv}(S)$ is simply

the closure of the convex hull, and the convex hull of any compact set is closed. If S is compact we have $z_S^* = \inf\{c^T x \mid x \in \text{clconv}(S)\}$ (see [83]). Thus for lower bounding it is sufficient to consider convex relaxations.

Convexity is still not sufficient for tractability; as noted earlier convex problems can be NP-hard. With MILP there is an important representation theorem due to Meyer [109]. Consider the mixed-integer set $M := \{(x, y) \mid Ax + Gy \leq b, x \in \mathbb{R}^k, y \in \mathbb{Z}^\ell\}$, where A, G, b are rational matrices of appropriate dimension. The convex hull of M can be represented with a rational polyhedra, i.e. there exist rational A', G', b' such that $\text{conv}(M) = \{A'x + G'y \leq b'\}$. For a broad class of polynomial programs [120], i.e. f, g , polynomial and $\Omega = \mathbb{R}^n$, the convex hull of the feasible region can be represented using additional variables with SDP via the Lasserre hierarchy [91]. However, representability is not sufficient for tractability, as there may be a large number of constraints and/or variables involved. Furthermore, there may not be an efficient procedure to generate the hull. For instance, with MILP one cannot in general represent the convex hull of M with a polynomial-sized linear program (e.g. [52, 137]).

A saving grace is that a full description of a convex hull is not required to optimize over it. We can, for instance, solve a relaxation and obtain a solution $x_R^* \in R$, and apply a separation algorithm over $\text{conv}(S)$. If $x_R^* \notin S$, then the separation algorithm returns a separating cut: a valid inequality $g(x) \leq 0$ such that $g(x_R^*) > 0$. The cut is added to the relaxation and the procedure is iterated — this is known as a cutting plane algorithm. Kelley [78] introduced linear cuts for convex constraints, thereby solving a convex problem using linear programming and a cutting plane algorithm. Later analysis provided an important characterization of this approach: a convex problem can be solved to ϵ -optimality if and only if linear separation can be done at fixed accuracy in polynomial time (see Thm 3.1 in [64]). One example of a problem with exponential-sized description but polynomial separation is the minimization of submodular functions [50, 104].

Cuts can be applied at any node of the branch-and-bound tree, and for MILP a host of cuts are available (e.g. [28]). There is a tradeoff between the breadth of applicable problems, the difficulty of generating cuts, and the strength of cuts. Furthermore, there is a choice of where to apply cuts in the tree, how many to add, and whether to purge cuts that are not binding in subproblems. For instance, Gomory's cutting plane algorithm [60] will solve pure integer linear programs in finite time, so it is a global algorithm that can be hybridized with branch-and-bound to form branch-and-cut. Cuts for mixed-integer convex problems have been developed more recently, so there are fewer results than in MILP (e.g. [12, 35]). Cuts for continuous nonconvex nonlinear constraints (e.g. [25, 142]) are rarer still; work has mostly focused on cuts for convex envelopes (e.g. [156, 155]).

1.5 Chapter Summaries

Chapter 2 describes a spatial branch-and-cut approach for nonconvex Quadratically Constrained Quadratic Programs with bounded complex variables (CQCQP). Linear valid inequalities are added at each node of the search tree to strengthen semidefinite programming relaxations of CQCQP. These valid inequalities are derived from the convex hull description of a nonconvex set of 2×2 positive semidefinite Hermitian matrices subject to rank-one constraint. Branching rules are proposed based on an alternative to a rank-one constraint that allows for local measurement of constraint violation. The algorithm is applied to solve the Alternating Current Optimal Power Flow problem with complex variables and the Box-constrained Quadratic Programming problem with real variables.

Chapter 3 builds on Chapter 2 with focus on the Alternating Current Optimal Power Flow (ACOPF) problem. ACOPF may be solved to global optimality with a semidefinite programming (SDP) relaxation in cases where its QCQP formulation attains zero duality gap. However, when there is positive duality gap, no optimal solution to the SDP relaxation is feasible for ACOPF. One way to find a global optimum is to partition the problem using a spatial branch-and-bound method. Tightening upper and lower variable bounds can improve solution times in spatial branching by potentially reducing the number of partitions needed. This chapter proposes special-purpose closed-form bound tightening methods to tighten limits on nodal powers, line flows, phase angle differences, and voltage magnitudes. Variants of IEEE test cases with high duality gaps are constructed in order to demonstrate the effectiveness of the bound tightening procedures.

In Chapter 4 we develop second-order cone cuts for a positive semidefinite constraint on a Hermitian matrix. These can be implemented in a sparse manner, which is important for applicability to the SDP relaxation of ACOPF. We provide an in-depth discussion on sparsity in SDP relaxations. These cuts allow one to trade-off the strength of the SDP relaxation with the computational expense. Second-order cone cuts are better able to capture the nonlinearity of the positive semidefinite constraint compared to standard outer-approximating linear cuts. Moreover, commercial solvers can solve mixed-integer conic problems, so this has the potential to be used to solve mixed-integer problems with power flow constraints.

Chapter 5 is on the Unit Commitment problem, which is an important scheduling problem in power systems operations. Recent advances in mathematical programming have significantly improved the tractability of Mixed Integer Second-Order Cone Programming (MISOCP). In this chapter we leverage MISOCP to solve Unit Commitment problem with quadratic costs. We show that conic strengthening can improve solution times by strengthening the relaxation.

In Chapter 6 we consider Mixed-Integer Geometric Programming (MIGP), which is an integer programming problem with convex constraints that involve the exponential function. It can be used to model a variety of physics and engineering problems, as well as risk constraints and exponential utility. A generic approach to solving MIGP is to treat it as a mixed-integer convex problem and apply linear outer-approximation techniques. We consider two additional approaches that are applicable to bounded MIGP and compare them with the benchmark linear outer-approximation. First, we consider conic outer-approximation of an exponential function with bounded domain. Second, we apply cuts that exploit submodular structure. We test our approach on three applications: a capital budgeting problem, the classic system reliability redundancy problem, and a logistic regression problem with cardinality.

Chapter 7 concludes the dissertation and presents future research directions.

Chapter 2

Spatial Branch-and-Cut for Complex Bounded QCQP

2.1 Introduction

The nonconvex quadratically-constrained quadratic programming problem with complex bounded variables (CQCQP) has numerous applications in signal processing [162, 46, 71] and control theory [22], among others. Our main motivation for developing an algorithm for CQCQP is to solve power flow problems with alternating current [92, 73]. We consider the following formulation of CQCQP:

$$\begin{aligned}
 & \min x^* Q_0 x + \operatorname{Re}(c_0^* x) + b_0 \\
 \text{(CQCQP)} \quad & \text{s.t. } x^* Q_i x + \operatorname{Re}(c_i^* x) + b_i \leq 0, \quad i = 1, \dots, m \\
 & \ell \leq x \leq u \\
 & x \in \mathbb{C}^n.
 \end{aligned}$$

We denote the conjugate transpose operator by $*$, and real components with $\operatorname{Re}(\cdot)$ and imaginary components with $\operatorname{Im}(\cdot)$. The decision vector $x \in \mathbb{C}^n$ has complex entries, and the remaining terms are data: Hermitian matrices $Q_i \in \mathbb{H}^{n \times n}$, real vector $b \in \mathbb{R}^n$, and complex vector $c_i \in \mathbb{C}^n$. Note that we need only assume that the magnitudes $|x_i|$ are bounded since this implies that the real and imaginary components of x are bounded. Variable bounds $\ell \leq x \leq u$ are component-wise inequalities in the complex space.

Real bounded QCQP (RQCQP) is a special case of CQCQP as all imaginary components can be set to 0 using the constraints $\operatorname{Re}(ix_i) = 0 \quad \forall i$, where $i := \sqrt{-1}$.

Note that CQCQP can be transformed into RQCQP by defining separate real decision vectors to represent $\text{Re}(x)$ and $\text{Im}(x)$. However, we consider the CQCQP formulation as presented in order to exploit its particular structure, from which we derive valid inequalities and spatial branching rules.

In this chapter we give a spatial branch-and-cut (SBC) approach to solve CQCQP. For brevity we assume familiarity with the general spatial branching framework; the reader is referred to Belotti et al. [19] for a thorough treatment on the subject. There are several spatial branching algorithms exploiting the structure of RQCQP (e.g. [99, 110, 135, 17, 130]); however, to the best of our knowledge we present the first spatial branching algorithm developed specifically for CQCQP. The implementation has three distinguishing features. First, we use a complex SDP formulation strengthened with valid inequalities. These inequalities are derived from the description of the convex hull of a nonconvex set of 2×2 positive semidefinite Hermitian matrices subject to rank-one constraint that arises from a lifted formulation of CQCQP. Second, we propose branching on the entries of the relaxation's decision matrix, and we consider branching rules based on an alternative measure of constraint violation in lieu of a matrix rank constraint. Third, we develop bound tightening procedures based on closed-form solutions.

Computational experiments are conducted on the Alternating Current Optimal Power Flow (ACOPF) problem and the Box-constrained Quadratic Programming (BoxQP) problem. ACOPF is a generation dispatch problem that models alternating current (AC) using steady-state power flow equations, which are nonconvex quadratic constraints that relate power and voltage at buses and across transmission lines. ACOPF is commonly solved with iterative Newton-type solvers (e.g. [153]). ACOPF may be modeled as a CQCQP problem, and there has been a recent interest in solving this formulation with a SDP relaxation (e.g. [16, 92]) and branch-and-bound methods (e.g. [130, 61, 85]) due to the potential for establishing global optimality. BoxQP is a well-studied nonconvex quadratic programming problem with nonhomogeneous quadratic objective and bounded real variables. BoxQP can be solved with a problem-specific finite branch-and-bound method [34]; therefore, it provides a useful benchmark for the more general approach presented here.

The rest of the chapter is organized as follows: Section 2.2 details the spatial branch-and-cut algorithm with three major components: valid inequalities from the convex hull description of rank-one constrained relaxations, branching on complex entries, and bound tightening procedures; Section 2.3 contains results from computational experiments with ACOPF and BoxQP problems; Section 2.4 concludes the chapter.

2.2 Spatial Branch-and-Cut

The convex relaxation of QCQP we consider comes from the rank-one constraint from a standard SDP reformulation [105, 146]:

$$\min \langle Q_0, X \rangle + \operatorname{Re}(c_0^* x) + b_0 \quad (2.1)$$

$$\text{(CSDP)} \quad \text{s.t.} \quad \langle Q_i, X \rangle + \operatorname{Re}(c_i^* x) + b_i \leq 0, \quad i = 1, \dots, m \quad (2.2)$$

$$\ell \leq x \leq u \quad (2.3)$$

$$\begin{bmatrix} 1 & x^* \\ x & X \end{bmatrix} \succeq 0, \quad (2.4)$$

where $X \in \mathbb{H}^{n \times n}$ is a Hermitian submatrix of decision variables. Imposing a rank-one constraint on the matrix $Y := \begin{bmatrix} 1 & x^* \\ x & X \end{bmatrix}$ gives an equivalent reformulation of QCQP.

A desirable property for a relaxation is exactness given sufficient branching. For instance, for bounded Mixed-Integer Linear Programming problems, a search tree created by integer variable branching has finite depth because the relaxation is exact when all integer variables are fixed. CSDP suffers from the fact that even when all entries of $\operatorname{diag}(X)$ and x have fixed values, the rank-one constraint may not be satisfied; we will present an example in the next subsection. Therefore, we shall strengthen CSDP with valid inequalities that ensure the relaxation is exact when variables are fixed.

This section is divided into three subsections. First, we present some examples of nonconvergence when branching with the CSDP relaxation. Second, we derive valid inequalities to strengthen the SDP relaxation. Third, we propose a methodology for branching on the entries of matrix Y .

Nonconvergence Example

Consider the following constraints defining a real QCQP with one variable:

$$\begin{aligned} \ell &\leq w \leq u \\ w &= w^2 \end{aligned}$$

By inspection, the problem is infeasible for any $u < 1$. The SDP relaxation is:

$$\begin{aligned} 0 &\leq w \leq u \\ w &= W \\ w^2 &\leq W \end{aligned}$$

This is equivalent to the nonempty feasible constraint set:

$$\begin{aligned} w &= W \\ 0 &\leq W \leq u \end{aligned}$$

Let us also consider an example specific to ACOPF. In the example we have that even for fixed voltage magnitudes, generation, and bus angle difference, the SDP is feasible while the primal problem is not. Consider a two-bus system with zero resistance and reactance $x = 0.1pu$. Furthermore, let $Y = jB$, where $B_{11} = B_{22} = -B_{12} = -10$. The ACOPF instance is:

$$\begin{aligned} |V_1| &= |V_2| = 1 \\ \theta_{12} = 0 &\iff \text{Im}(V_1)\text{Re}(V_2) = \text{Re}(V_1)\text{Im}(V_2) \\ P_1 &= P_2 = 0 \\ Q_1 &= Q_2 = -10 \\ P_1 &= B_{12}[\text{Im}(V_1)\text{Re}(V_2) - \text{Re}(V_1)\text{Im}(V_2)] \\ P_2 &= B_{12}[\text{Im}(V_2)\text{Re}(V_1) - \text{Re}(V_2)\text{Im}(V_1)] \\ Q_1 &= -B_{11}|V_1|^2 - B_{12}[\text{Re}(V_1)\text{Re}(V_2) + \text{Im}(V_1)\text{Im}(V_2)] \\ Q_2 &= -B_{22}|V_2|^2 - B_{12}[\text{Re}(V_1)\text{Re}(V_2) + \text{Im}(V_1)\text{Im}(V_2)] \end{aligned}$$

This is infeasible since $Q_1 = -B_{11}|V_1|^2 = -10$, but $B_{12}[\text{Re}(V_1)\text{Re}(V_2) + \text{Im}(V_1)\text{Im}(V_2)]$ attains a nonzero value. However, the SDP relaxation is:

$$\begin{aligned} W_{11} &= W_{22} = 1 \\ T_{12} &= 0 \\ P_1 &= P_2 = 0 \\ Q_1 &= Q_2 = -10 \\ P_1 &= B_{12}T_{12} \\ P_2 &= -B_{12}T_{12} \\ Q_1 &= -B_{11}W_{11} - B_{12}W_{12} \\ Q_2 &= -B_{22}W_{22} - B_{12}W_{12} \\ W_{12}^2 + T_{12}^2 &\leq W_{11}W_{22} \end{aligned}$$

This has one feasible solution at $W_{12} = 0$, hence a spatial branching procedure with variable branching cannot prove infeasibility on this problem using the standard SDP relaxation.

Valid Inequalities

In this section we describe the convex hull of a nonconvex relaxation of CQCQP derived from 2×2 positive semidefinite Hermitian matrices subject to the rank-one constraint. Let \mathcal{J}_C be the set of Hermitian matrices $X := W + iT \in \mathbb{H}^{2 \times 2}$ satisfying the following constraints:

$$L_{11} \leq W_{11} \leq U_{11}, \quad (2.5a)$$

$$L_{22} \leq W_{22} \leq U_{22}, \quad (2.5b)$$

$$L_{12}W_{12} \leq T_{12} \leq U_{12}W_{12}, \quad (2.5c)$$

$$W_{11}W_{22} = W_{12}^2 + T_{12}^2. \quad (2.5d)$$

Since $W + iT$ is Hermitian, we have $W_{21} = W_{12}, T_{12} = -T_{21}$. Therefore it suffices to consider the variables $W_{11}, W_{22}, T_{12}, W_{12}$. Valid inequalities (2.5a)–(2.5c) can be derived from any instance of CQCQP and added to CSDP for all 2×2 principal minors of Y (see Appendix A.1). U and L are matrices of upper and lower bounds, so $L_{ij} \leq U_{ij}, \forall i, j \in \{1, 2\}$. Moreover, we will assume the diagonal elements of X have at least trivial nonnegative lower bounds, so $L_{11}, L_{22} \geq 0$. Due to the nonnegativity assumption, constraint (2.5d) is equivalent in the space of X to the constraint $X = xx^*$, since $\{xx^* | x \in \mathbb{C}^n\}$ is the set of rank one matrices together with the rank zero matrix.

In this section we will give a description of the convex hull of \mathcal{J}_C . Let us first establish valid inequalities for \mathcal{J}_C . To do so, it will be convenient to use the following sigmoid function:

$$f(x) := \begin{cases} (\sqrt{1+x^2} - 1)/x, & x \neq 0 \\ 0, & x = 0 \end{cases}.$$

Remark 1. $f(x)$ is increasing, strictly bounded above by +1 and strictly bounded below by -1.

Consider the following linear inequalities:

$$\pi_0 + \pi_1 W_{11} + \pi_2 W_{22} + \pi_3 W_{12} + \pi_4 T_{12} \geq U_{22}W_{11} + U_{11}W_{22} - U_{11}U_{22}, \quad (2.6a)$$

$$\pi_0 + \pi_1 W_{11} + \pi_2 W_{22} + \pi_3 W_{12} + \pi_4 T_{12} \geq L_{22}W_{11} + L_{11}W_{22} - L_{11}L_{22}, \quad (2.6b)$$

where the coefficients π are defined as

$$\begin{aligned}\pi_0 &:= -\sqrt{L_{11}L_{22}U_{11}U_{22}}, \\ \pi_1 &:= -\sqrt{L_{22}U_{22}}, \\ \pi_2 &:= -\sqrt{L_{11}U_{11}}, \\ \pi_3 &:= (\sqrt{L_{11}} + \sqrt{U_{11}})(\sqrt{L_{22}} + \sqrt{U_{22}}) \frac{1 - f(L_{12})f(U_{12})}{1 + f(L_{12})f(U_{12})}, \\ \pi_4 &:= (\sqrt{L_{11}} + \sqrt{U_{11}})(\sqrt{L_{22}} + \sqrt{U_{22}}) \frac{f(L_{12}) + f(U_{12})}{1 + f(L_{12})f(U_{12})}.\end{aligned}$$

Lemma 2.2.1. *For $\alpha_{12} \in \{L_{12}, U_{12}\}$ we have*

$$1 - f(L_{12})f(U_{12}) + \alpha_{12}(f(L_{12}) + f(U_{12})) = (1 + f(L_{12})f(U_{12}))\sqrt{1 + \alpha_{12}^2}. \quad (2.7)$$

Proof. If $\alpha_{12} = 0$, then equality (2.7) follows immediately. Otherwise, suppose $\alpha_{12} = L_{12} \neq 0$. Then we have

$$\begin{aligned}& 1 - f(L_{12})f(U_{12}) + L_{12}(f(L_{12}) + f(U_{12})) - (1 + f(L_{12})f(U_{12}))\sqrt{1 + L_{12}^2} \\ &= -f(L_{12})f(U_{12}) + L_{12}f(U_{12}) - f(L_{12})f(U_{12})\sqrt{1 + L_{12}^2} \\ &= f(U_{12})[L_{12} - f(L_{12})(1 + \sqrt{1 + L_{12}^2})] \\ &= f(U_{12})[L_{12} - L_{12}^2/L_{12}] \\ &= 0.\end{aligned}$$

Therefore

$$1 - f(L_{12})f(U_{12}) + L_{12}(f(L_{12}) + f(U_{12})) = (1 + f(L_{12})f(U_{12}))\sqrt{1 + L_{12}^2}.$$

The case where $\alpha_{12} = U_{12} \neq 0$ follows by symmetry. \square

Proposition 2.2.2. *Inequalities (2.6a) and (2.6b) are valid for \mathcal{J}_C .*

Proof. For any convex function f , the secant line connecting $(a, f(a))$ and $(b, f(b))$ lies above the graph of f . Thus $\ell \leq x \leq u \implies (\ell + u)x - \ell u \geq x^2$. Now for $k \in \{1, 2\}$, since $\sqrt{L_{kk}} \leq \sqrt{W_{kk}} \leq \sqrt{U_{kk}}$, applying this secant principle yields

$$(\sqrt{L_{kk}} + \sqrt{U_{kk}})\sqrt{W_{kk}} \geq \sqrt{L_{kk}U_{kk}} + W_{kk}. \quad (2.8a)$$

Multiplying inequalities from (2.8a) for $k = 1$ and $k = 2$ gives

$$\sqrt{W_{11}W_{22}}(\sqrt{L_{11}} + \sqrt{U_{11}})(\sqrt{L_{22}} + \sqrt{U_{22}}) \geq (\sqrt{L_{11}U_{11}} + W_{11})(\sqrt{L_{22}U_{22}} + W_{22}).$$

Rearranging terms, we have

$$\sqrt{W_{11}W_{22}}(\sqrt{L_{11}} + \sqrt{U_{11}})(\sqrt{L_{22}} + \sqrt{U_{22}}) + \pi_0 + \pi_1W_{11} + \pi_2W_{22} \geq W_{11}W_{22}. \quad (2.8b)$$

The right-hand-sides of constraints (2.6a) and (2.6b) are RLT inequalities [10] – and thus are valid – for the bilinear term $W_{11}W_{22}$, which is the right-hand-side of the valid inequality (2.8b). It remains to show that the left-hand-side of (2.8b) is overestimated by the left-hand sides of (2.6a) and (2.6b), i.e.:

$$\pi_3W_{12} + \pi_4T_{12} \geq \sqrt{W_{11}W_{22}}(\sqrt{L_{11}} + \sqrt{U_{11}})(\sqrt{L_{22}} + \sqrt{U_{22}}). \quad (2.8c)$$

Let $T_{12} = \alpha W_{12}$ for some $L_{12} \leq \alpha_{12} \leq U_{12}$. Note that constraint (2.5c) restricts W_{12} to be nonnegative, so from constraints (2.5c) we have:

$$\sqrt{W_{11}W_{22}} = \sqrt{1 + \alpha_{12}^2}W_{12}. \quad (2.8d)$$

Substituting equality (2.8d) into inequality (2.8c), we therefore want to show that

$$(\pi_3 + \alpha_{12}\pi_4)W_{12} \geq \sqrt{1 + \alpha_{12}^2}(\sqrt{L_{11}} + \sqrt{U_{11}})(\sqrt{L_{22}} + \sqrt{U_{22}})W_{12} \quad (2.8e)$$

for any $L_{12} \leq \alpha_{12} \leq U_{12}$. Replacing the coefficients with their definitions and simplifying, we get the following equivalent condition:

$$1 - f(L_{12})f(U_{12}) + \alpha_{12}(f(L_{12}) + f(U_{12})) - (1 + f(L_{12})f(U_{12}))\sqrt{1 + \alpha_{12}^2} \geq 0. \quad (2.8f)$$

To show that inequality (2.8f) is valid, first observe that the second derivative of the left-hand side with respect to α is $-(1 + f(L_{12})f(U_{12}))/(1 + \alpha^2)^{3/2}$. Thus the left-hand side is concave w.r.t α (as noted in Remark 1 we have that $|f(L_{12})|, |f(U_{12})| < 1$). Therefore, we only need to check that the left-hand side is nonnegative for $\alpha \in \{L_{12}, U_{12}\}$. From Lemma 2.2.1 we have that the left-hand side is exactly zero, and so inequality (2.8f) and consequently inequalities (2.6a) and (2.6b) are valid for \mathcal{J}_C . \square

Let \mathcal{J}_S be the set of feasible solutions to the standard SDP relaxation of \mathcal{J}_C :

$$L_{11} \leq W_{11} \leq U_{11}, \quad (2.9a)$$

$$L_{22} \leq W_{22} \leq U_{22}, \quad (2.9b)$$

$$L_{12}W_{12} \leq T_{12} \leq U_{12}W_{12} \quad (2.9c)$$

$$W_{11}W_{22} \geq W_{12}^2 + T_{12}^2. \quad (2.9d)$$

Remark 2. For $n = 2$, $W + iT \succeq 0$ is equivalent to the principal minor constraints: $W_{11} \geq 0, W_{22} \geq 0, W_{11}W_{22} \geq W_{12}^2 + T_{12}^2$. Since by assumption $L_{11}, L_{22} \geq 0$, then constraint (2.9d) is equivalent to the positive semidefinite constraint for \mathcal{J}_S .

Let \mathcal{J}_V be the set of X satisfying inequalities (2.6a)-(2.6b). We shall prove that the convex hull of \mathcal{J}_C can be obtained by adding the valid inequalities (2.6a)-(2.6b) to the SDP relaxation.

Proposition 2.2.3. $\text{conv}(\mathcal{J}_C) = \mathcal{J}_S \cap \mathcal{J}_V$.

Proof. From Lemma 2.2.2 we have that $\text{conv}(\mathcal{J}_C) \subseteq \mathcal{J}_S \cap \mathcal{J}_V$. From constraints (2.5a)-(2.5b) and (2.5d) we can observe that $\mathcal{J}_S \cap \mathcal{J}_V$ is bounded. Thus to prove that $\mathcal{J}_S \cap \mathcal{J}_V \subseteq \text{conv}(\mathcal{J}_C)$, it is sufficient to ensure that all the extreme points of $\mathcal{J}_S \cap \mathcal{J}_V$ are in \mathcal{J}_C . First, let us invoke the following claim:

Claim 1. If $W_{12} = 0$, then either $X \notin \mathcal{J}_S \cap \mathcal{J}_V$ or $X \in \text{conv}(\mathcal{J}_C)$.

Proof. By way of contradiction, suppose there exists $\bar{X} \in \mathcal{J}_S \cap \mathcal{J}_V \setminus \text{conv}(\mathcal{J}_C)$ such that $\bar{W}_{12} = 0$. From constraint (2.5c) we have that $\bar{T}_{12} = 0$. Consequently, if $\bar{W}_{11} = 0$ or $\bar{W}_{22} = 0$, then constraint (2.5d) is satisfied, so either $\bar{X} \notin \mathcal{J}_S \cap \mathcal{J}_V$ or $\bar{X} \in \text{conv}(\mathcal{J}_C)$.

By Claim 2.2.2, throughout the proof, we may assume that $\bar{W}_{11}\bar{W}_{22} > 0$, which $\bar{W}_{11}\bar{W}_{22} > 0$ implies that the right-hand side of constraint (2.6b) is nonnegative since $0 \leq L_{kk} \leq \bar{W}_{kk}$, $k \in \{1, 2\}$, and so $L_{22}\bar{W}_{11} + L_{11}\bar{W}_{22} - L_{11}L_{22} \geq L_{11}\bar{W}_{22} \geq 0$. On the left-hand side, all terms are nonpositive, so we require all terms to be zero in order to ensure $\bar{X} \in \mathcal{J}_S \cap \mathcal{J}_V$. Since we have that $\bar{W}_{11}\bar{W}_{22} > 0$, then $\pi_1\bar{W}_{11} = \pi_2\bar{W}_{22} = 0$ only if $L_{11} = L_{22} = 0$.

Now let us define the following two matrices:

$$U^1 := \begin{bmatrix} U_{11} & 0 \\ 0 & 0 \end{bmatrix}, U^2 := \begin{bmatrix} 0 & 0 \\ 0 & U_{22} \end{bmatrix}.$$

Since $L_{11} = L_{22} = 0$, then $U^1, U^2 \in \mathcal{J}_C$, and since $\bar{W}_{12} = \bar{T}_{12} = 0$. Checking constraint (2.6a) we have $0 \geq U_{22}W_{11} + U_{11}W_{22} - U_{11}U_{22}$, so $\bar{X} \in \mathcal{J}_S \cap \mathcal{J}_V$ provided $W_{11} \leq U_{11}, W_{22} \leq U_{22}$. Observe that \bar{X} can be expressed as the convex combination of U^1, U^2 , and the zeros matrix. $L_{11} = L_{22} = 0$ implies the zeros matrix is a member of \mathcal{J}_C , which in turn implies $\bar{X} \in \text{conv}(\mathcal{J}_C)$, which contradicts our initial assumption. \square

Now we will prove that $\text{ext}(\mathcal{J}_S \cap \mathcal{J}_V) \in \mathcal{J}_C$. Observe that if the constraint (2.9d) is binding at an extreme point of $\mathcal{J}_S \cap \mathcal{J}_V$, then it is a member of \mathcal{J}_C . Moreover, by Claim (1), if a point \bar{X} with $\bar{W}_{12} = 0$ is in $\text{ext}(\mathcal{J}_S \cap \mathcal{J}_V)$, then $\bar{X} \in \text{conv}(\mathcal{J}_C)$. It follows that \bar{X} is a member of \mathcal{J}_C since $\mathcal{J}_S \cap \mathcal{J}_V \supseteq \text{conv}(\mathcal{J}_C) \supseteq \mathcal{J}_C$. Therefore, we shall check by cases for any extreme point where constraint (2.9d) is not binding and $W_{12} \neq 0$.

Case 1: Constraints (2.6a) and (2.6b) are not binding:

If constraint (2.9d) is not binding, then we require at least four linearly independent linear constraints to be binding. To obtain four such constraints, we require the two variable bounds (2.5a) and (2.5b) to be binding. Moreover, we require that constraint (2.5c) is binding on both sides with $L_{12} \neq U_{12}$, which implies $W_{12} = T_{12} = 0$. By Claim (1), this point can be disregarded.

Case 2: Constraints (2.6a) and (2.6b) are both binding:

Since constraints (2.6a) and (2.6b) share the same coefficients π_3, π_4 for W_{12}, T_{12} , then for an extreme point we require that constraint (2.5c) is binding on at least one side. Due to Claim (1) we need only consider $W_{12} \neq 0$, so constraint (2.5c) can count for at most one linearly independent constraint; thus, let $T_{12} = \alpha_{12}W_{12}$, where $\alpha_{12} \in \{L_{12}, U_{12}\}$. This gives at most three linearly independent constraints, (2.5c), (2.6a) and (2.6b), so at least one of the variable bounds (2.5a) and (2.5b) must be binding. Define α_{kk} so that $W_{kk} = (U_{kk} - L_{kk})\alpha_{kk} + L_{kk}$ for $k \in \{1, 2\}$. Since at

least one of W_{11}, W_{22} is at a variable bound, then $\alpha_{kk} \in \{0, 1\}$ for either $k = 1$ or $k = 2$. Moreover, the right-hand sides of constraints (2.6a) and (2.6b) must be equal since the left-hand sides are the same and both constraints are binding in this case. Therefore, we can write:

$$\begin{aligned} & U_{22}W_{11} + U_{11}W_{22} - U_{11}U_{22} = L_{22}W_{11} + L_{11}W_{22} - L_{11}L_{22} \\ \iff & (\alpha_{11} + \alpha_{22})[U_{11}U_{22} - L_{11}U_{22} - L_{22}U_{11} + L_{11}L_{22}] \\ & = U_{11}U_{22} - L_{11}U_{22} - L_{22}U_{11} + L_{11}L_{22} \\ \iff & \alpha_{11} + \alpha_{22} = 1. \end{aligned}$$

Thus we have either $\alpha_{11} = 1$, in which case $W_{11} = U_{11}, W_{22} = L_{22}$ or $\alpha_{22} = 1$, in which case $W_{11} = L_{11}, W_{22} = U_{22}$. First suppose $W_{11} = U_{11}, W_{22} = L_{22}$. Define the following matrix:

$$X^A := \begin{bmatrix} U_{11} & \sqrt{\frac{U_{11}L_{22}}{1+\alpha_{12}^2}} \\ \sqrt{\frac{U_{11}L_{22}}{1+\alpha_{12}^2}} & L_{22} \end{bmatrix} + i \begin{bmatrix} 0 & \alpha_{12}\sqrt{\frac{U_{11}L_{22}}{1+\alpha_{12}^2}} \\ -\alpha_{12}\sqrt{\frac{U_{11}L_{22}}{1+\alpha_{12}^2}} & 0 \end{bmatrix},$$

and as usual, denote the components as $X^A := W^A + iT^A$. By construction, we have that $(W_{12}^A)^2 + (T_{12}^A)^2 = W_{11}^A W_{22}^A$, so $X^A \in \mathcal{J}_C$. We want to show that, with X^A , constraints (2.5a)-(2.5c), (2.6a), and (2.6b) are binding. Constraints (2.5a)-(2.5c) can be confirmed by observation. Recall that constraints (2.6a) and (2.6b) share the same right-hand side:

$$U_{22}W_{11}^A + U_{11}W_{22}^A - U_{11}U_{22} = U_{11}L_{22}.$$

The left-hand side of constraint (2.6a) or (2.6b) is:

$$\begin{aligned}
 & \pi_0 + \pi_1 W_{11}^A + \pi_2 W_{22}^A + \pi_3 W_{12}^A + \pi_4 T_{12}^A \\
 = & \pi_0 + \pi_1 W_{11}^A + \pi_2 W_{22}^A + (\pi_3 + \alpha_{12}\pi_4)W_{12}^A \\
 = & -\sqrt{L_{11}L_{22}U_{11}U_{22}} - \sqrt{L_{22}U_{22}U_{11}} - \sqrt{L_{11}U_{11}L_{22}} \\
 & + (\sqrt{L_{11}} + \sqrt{U_{11}})(\sqrt{L_{22}} + \sqrt{U_{22}}) \\
 & \times \frac{1 - f(L_{12})f(U_{12}) + \alpha_{12}(f(L_{12}) + f(U_{12}))}{1 + f(L_{12})f(U_{12})} \sqrt{\frac{U_{11}L_{22}}{1 + \alpha_{12}^2}} \\
 = & \sqrt{U_{11}L_{22}} \\
 & \times \left(-\sqrt{L_{11}U_{22}} - \sqrt{U_{11}U_{22}} - \sqrt{L_{11}L_{22}} \right. \\
 & \left. + (\sqrt{L_{11}} + \sqrt{U_{11}})(\sqrt{L_{22}} + \sqrt{U_{22}}) \frac{1 - f(L_{12})f(U_{12}) + \alpha_{12}(f(L_{12}) + f(U_{12}))}{(1 + f(L_{12})f(U_{12}))\sqrt{1 + \alpha_{12}^2}} \right) \\
 = & U_{11}L_{22}.
 \end{aligned}$$

The last equality follows from Lemma 2.2.1. The argument for the case $W_{11} = L_{11}, W_{22} = U_{22}$ follows by symmetry.

Case 3: Exactly one of the constraints (2.6a) and (2.6b) is binding:

From Claim (1) we need only consider $W_{12} \neq 0$, so constraint (2.5c) can count for at most one linearly independent constraint; thus, let $T_{12} = \alpha_{12}W_{12}$, where $\alpha_{12} \in \{L_{12}, U_{12}\}$. This gives us at most two linearly independent linear constraints: (2.5c) and either (2.6a) or (2.6b). Thus both variable bounds (2.5a) and (2.5b) must be binding. In Case 2 we have already considered the possibilities that $W_{11} = U_{11}, W_{22} = L_{22}$ or $W_{11} = U_{11}, W_{22} = L_{22}$. Suppose, then, that $W_{11} = U_{11}, W_{22} = U_{22}$ and define the corresponding matrix:

$$X^B := \begin{bmatrix} U_{11} & \sqrt{\frac{U_{11}U_{22}}{1+\alpha_{12}^2}} \\ \sqrt{\frac{U_{11}U_{22}}{1+\alpha_{12}^2}} & U_{22} \end{bmatrix} + i \begin{bmatrix} 0 & \alpha_{12}\sqrt{\frac{U_{11}U_{22}}{1+\alpha_{12}^2}} \\ -\alpha_{12}\sqrt{\frac{U_{11}U_{22}}{1+\alpha_{12}^2}} & 0 \end{bmatrix},$$

and as usual, denote the components as $X^B := W^B + iT^B$. By construction, we have that $(W_{12}^B)^2 + (T_{12}^B)^2 = (W_{11}^B W_{22}^B)$, so $X^B \in \mathcal{J}_C$. We want to show that, with X^B , constraints (2.5a)-(2.5c), and (2.6a) are binding. This can be done as in Case 2, by

invoking Lemma 2.2.1:

$$\begin{aligned}
 & \pi_0 + \pi_1 W_{11}^B + \pi_2 W_{22}^B + \pi_3 W_{12}^B + \pi_4 T_{12}^B \\
 = & \pi_0 + \pi_1 W_{11}^B + \pi_2 W_{22}^B + (\pi_3 + \alpha_{12}\pi_4)W_{12}^B \\
 = & -\sqrt{L_{11}L_{22}U_{11}U_{22}} - \sqrt{L_{22}U_{22}U_{11}} - \sqrt{L_{11}U_{11}U_{22}} \\
 & + (\sqrt{L_{11}} + \sqrt{U_{11}})(\sqrt{L_{22}} + \sqrt{U_{22}}) \\
 & \times \frac{1 - f(L_{12})f(U_{12}) + \alpha_{12}(f(L_{12}) + f(U_{12}))}{1 + f(L_{12})f(U_{12})} \sqrt{\frac{U_{11}U_{22}}{1 + \alpha_{12}^2}} \\
 = & \sqrt{U_{11}U_{22}} \left(-\sqrt{L_{11}U_{22}} - \sqrt{U_{11}L_{22}} - \sqrt{L_{11}L_{22}} \right. \\
 & \left. + (\sqrt{L_{11}} + \sqrt{U_{11}})(\sqrt{L_{22}} + \sqrt{U_{22}}) \frac{1 - f(L_{12})f(U_{12}) + \alpha_{12}(f(L_{12}) + f(U_{12}))}{(1 + f(L_{12})f(U_{12}))\sqrt{1 + \alpha_{12}^2}} \right) \\
 = & U_{11}U_{22} \\
 = & U_{22}W_{11}^B + U_{11}W_{22}^B - U_{11}U_{22}.
 \end{aligned}$$

Finally, suppose that $W_{11} = L_{11}, W_{22} = L_{22}$ and define the corresponding matrix:

$$X^C := \begin{bmatrix} L_{11} & \sqrt{\frac{L_{11}L_{22}}{1+\alpha_{12}^2}} \\ \sqrt{\frac{L_{11}L_{22}}{1+\alpha_{12}^2}} & L_{22} \end{bmatrix} + \imath \begin{bmatrix} 0 & \alpha_{12}\sqrt{\frac{L_{11}L_{22}}{1+\alpha_{12}^2}} \\ -\alpha_{12}\sqrt{\frac{L_{11}L_{22}}{1+\alpha_{12}^2}} & 0 \end{bmatrix}.$$

By the same argument as used with X^B , which we omit to avoid repetition, X^C belongs to \mathcal{J}_C where constraints (2.5a)-(2.5c), and (2.6b) are binding. Thus, in all cases every extreme point belongs to \mathcal{J}_C . \square

The real case

We now consider the special case of the real variables. Let \mathcal{J}_R be the set of symmetric matrices W that satisfy the following constraints:

$$L_{11} \leq W_{11} \leq U_{11}, \quad (2.10a)$$

$$L_{22} \leq W_{22} \leq U_{22}, \quad (2.10b)$$

$$W_{11}W_{22} = W_{12}^2 \quad (2.10c)$$

\mathcal{J}_R is a special case of \mathcal{J}_C , and so we can obtain the following corollary.

Corollary 1. *The convex hull of \mathcal{J}_R can be described with the following constraints:*

$$L_{11} \leq W_{11} \leq U_{11} \quad (2.11a)$$

$$L_{22} \leq W_{22} \leq U_{22} \quad (2.11b)$$

$$\begin{aligned} (\sqrt{L_{11}} + \sqrt{U_{11}})(\sqrt{L_{22}} + \sqrt{U_{22}})W_{12} &\geq (U_{22} + \sqrt{L_{22}U_{22}})W_{11} \\ &+ (U_{11} + \sqrt{L_{11}U_{11}})W_{22} + \sqrt{L_{11}L_{22}U_{11}U_{22}} - U_{11}U_{22} \end{aligned} \quad (2.11c)$$

$$\begin{aligned} (\sqrt{L_{11}} + \sqrt{U_{11}})(\sqrt{L_{22}} + \sqrt{U_{22}})W_{12} &\geq (L_{22} + \sqrt{L_{22}U_{22}})W_{11} \\ &+ (L_{11} + \sqrt{L_{11}U_{11}})W_{22} + \sqrt{L_{11}L_{22}U_{11}U_{22}} - L_{11}L_{22} \end{aligned} \quad (2.11d)$$

$$W \succeq 0 \quad (2.11e)$$

Proof. This is a special case of Proposition 2.2.3 with $L_{12} = U_{12} = 0$, which due to constraint (2.5c) is equivalent to setting $T_{12} = 0$, resulting in a matrix with only real entries. \square

Numerical Example

Consider the following instance of \mathcal{J}_C :

$$W_{11}W_{22} = W_{12}^2 + T_{12}^2, \quad (2.12a)$$

$$0 \leq W_{11} \leq 1, \quad (2.12b)$$

$$1 \leq W_{22} \leq 4, \quad (2.12c)$$

$$0 \leq T_{12} \leq \frac{4}{3}W_{12}. \quad (2.12d)$$

The coefficients π of the valid inequalities (2.6a)-(2.6b) are

$$\begin{aligned} f(L_{12}) &= 0, f(U_{12}) = 0.5, \\ \pi_0 &= 0, \pi_1 = -2, \pi_2 = 0, \pi_3 = 3, \pi_4 = 1.5. \end{aligned}$$

Valid inequality (2.6a) is

$$-6W_{11} - W_{22} + 3W_{12} + 1.5T_{12} + 4 \geq 0,$$

and valid inequality (2.6b) is

$$-3W_{11} + 3W_{12} + 1.5T_{12} \geq 0.$$

The special case on reals, \mathcal{J}_R , is obtained by setting $L_{12} = U_{12} = 0$ and dropping T_{12} . For the real case, valid inequality (2.6a) is

$$-6W_{11} - W_{22} + 3W_{12} + 4 \geq 0,$$

and valid inequality (2.6b) is

$$-3W_{11} + 3W_{12} \geq 0.$$

The inequalities for the real case are shown in Figure 2.1. The feasible region of \mathcal{J}_R is depicted in the upper-left quadrant. The upper-right quadrant depicts the intersection of $W_{12}^2 = W_{11}W_{22}$ with the variable bounds at $W_{11} = 1$ and $W_{22} = 1$. Black spheres are centered around the intersection points of variable bounds on the cone. The lower-right quadrant depicts the cone with valid inequality (2.6a). The intersection is ellipsoidal and three of the highlighted points lie on the boundary of the valid inequality ($\{W_{11} = 0, W_{22} = 4, W_{12} = 0, \}, \{W_{11} = 1, W_{22} = 1, W_{12} = 1\}, \{W_{11} = 1, W_{22} = 4, W_{12} = 2\}$). The lower-left quadrant depicts the intersection of the cone with valid inequality (2.6b). The intersection is a hyperbola, and three highlighted points lie on the boundary of the valid inequality ($\{W_{11} = 0, W_{22} = 1, W_{12} = 0, \}, \{W_{11} = 0, W_{22} = 4, W_{12} = 0\}, \{W_{11} = 1, W_{22} = 1, W_{12} = 1\}$).

Comparison with RLT Inequalities

The complex valid inequalities are related to the RLT inequalities, which are valid inequalities for the set $\{(W, w) : W = ww', \ell \leq w \leq u\}$:

$$W_{ij} \leq u_j w_i + \ell_i w_j - \ell_i u_j, \quad (2.13a)$$

$$\text{(RLT)} \quad W_{ij} \leq \ell_j w_i + u_i w_j - \ell_j u_i, \quad (2.13b)$$

$$W_{ij} \geq \ell_j w_i + \ell_i w_j - \ell_i \ell_j, \quad (2.13c)$$

$$W_{ij} \geq u_j w_i + u_i w_j - u_i u_j. \quad (2.13d)$$

Inequalities (2.13a)–(2.13d) are derived from the RLT procedure of Serali and Adams [144]. They are also known as McCormick estimators as they can be derived from earlier work on convex envelopes by McCormick [108].

Define $Z := \begin{pmatrix} 1 \\ w \end{pmatrix} \begin{pmatrix} 1 \\ w \end{pmatrix}'$. Anstreicher and Burer [11] prove that $\text{conv}(\{Z : \ell \leq w \leq u, w \in \mathbb{R}^2\})$ is given by the RLT inequalities together with the SDP constraint $\begin{bmatrix} 1 & w \\ w & W \end{bmatrix} \succeq 0$. Consequently, $\text{conv}(\mathcal{J}_R)$ can be described with RLT inequalities using an extended formulation with the additional variables w_1, w_2 satisfying $\sqrt{L_{11}} \leq w_1 \leq \sqrt{U_{11}}, \sqrt{L_{22}} \leq w_2 \leq \sqrt{U_{22}}$. When $i = j$, the RLT inequalities are

$$W_{ii} \leq u_i w_i + \ell_i w_i - \ell_i u_i, \quad (2.14a)$$

$$W_{ii} \geq 2\ell_i w_i - \ell_i^2, \quad (2.14b)$$

$$W_{ii} \geq 2u_i w_i - u_i^2. \quad (2.14c)$$

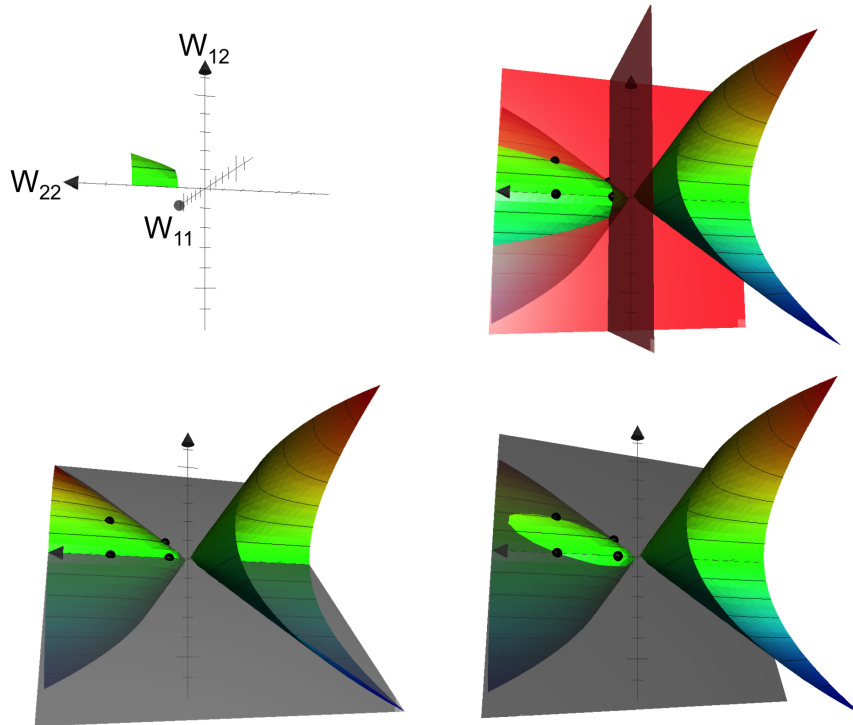


Figure 2.1: Clockwise, starting from top-left: \mathcal{J}_R ; cone of (2.12a), upper bound of (2.12b), and lower bound of (2.12c); cone of (2.12a) and valid inequality (2.6a); cone of (2.12a) and valid inequality (2.6b).

Inequality (2.14a) can also be obtained from (2.11c) or (2.11d) applied to $\begin{bmatrix} 1 & w_i \\ w_i & W_{ii} \end{bmatrix}$. Inequalities (2.14b)-(2.14c) are implied by the SDP constraint since $(w_i - ll_i)^2 \geq 0$, $(w_i - u_i)^2 \geq 0 \implies w_i^2 \geq \max\{2ll_i w_i - ll_i^2, 2u_i w_i - u_i^2\}$ and $W_{ii} \geq w_i^2$.

Note that for \mathcal{J}_C , one possible transformation to RQCQP involves a matrix of the form

$$\begin{pmatrix} 1 \\ w \\ t \end{pmatrix} \begin{pmatrix} 1 \\ w \\ t \end{pmatrix}',$$

where the components of the complex vector $x := w + it$ are treated as separate decision variables. In this case there is no guarantee that the RLT inequalities together with the SDP constraint (2.9d) will yield the convex hull of \mathcal{J}_C .

Numerical Example (cont.)

We continue with the numerical example of \mathcal{J}_C , and we will show that there is a point satisfying the positive semidefinite condition and the RLT inequalities, but is outside $\text{conv}(\mathcal{J}_C)$. Thus a standard application of RLT inequalities is insufficient to describe $\text{conv}(\mathcal{J}_C)$. Let us add the complex variables x_1, x_2 with the following bounds on magnitude:

$$\begin{aligned} 0 &\leq |x_{11}| \leq 1, \\ 1 &\leq |x_{22}| \leq 2. \end{aligned}$$

Considering the real and imaginary components of x as separate decision variables, let us transform the problem into RQCQP. Moreover, we require bounds on $\text{Re}(x), \text{Im}(x)$. We shall use the magnitude-implied bounds:

$$\begin{aligned} |w_1| &\leq 1, \quad |t_1| \leq 1, \\ |w_2| &\leq 2, \quad |t_2| \leq 2. \end{aligned}$$

Now $W_{ij} = \text{Re}(x_i x_j^*) = w_i w_j + t_i t_j$, and $T_{ij} = \text{Im}(x_i x_j^*) = t_i w_j - w_i t_j$. Applying RLT inequalities to each bilinear term, we obtain the following inequalities:

$$W_{12} \leq -|2w_1 - w_2| - |2t_1 - t_2| + 4 \quad (2.15a)$$

$$W_{12} \geq |2w_1 + w_2| + |2t_1 + t_2| - 4 \quad (2.15b)$$

$$T_{12} \leq -|2t_1 - w_2| - |2w_1 + t_2| + 4 \quad (2.15c)$$

$$T_{12} \geq |2t_1 + w_2| + |2w_1 - t_2| - 4 \quad (2.15d)$$

$$W_{11} \leq 2, \quad (2.15e)$$

$$W_{11} \geq 2|w_1| + 2|t_1| - 2, \quad (2.15f)$$

$$W_{22} \leq 8, \quad (2.15g)$$

$$W_{22} \geq 4|w_1| + 4|t_1| - 8. \quad (2.15h)$$

RLT inequalities (2.15) admit the solution $x = 0, T_{12} = 0, W_{12} = 0, W_{11} = 1, W_{22} = 4$, which satisfies constraints (2.12b)-(2.12d) and the positive semidefinite constraint. However, this solution violates valid inequality (2.6b), where $W_{12} = T_{12} = 0$ implying $W_{11} = 0$.

Now suppose we can strengthen the RLT inequalities with some given bounds on the components of x :

$$\begin{aligned} 0 &\leq w_1 \leq 1, \quad t_1 = 0, \\ 0.6 &\leq w_2 \leq 2, \quad -1.6 \leq t_2 \leq 0. \end{aligned}$$

The bounds are constructed by fixing t_1 , and tightening the other variable bounds as much as possible without excluding solutions to \mathcal{J}_C specified by (2.12). The RLT inequalities are:

$$W_{12} \leq \min\{2w_1, 0.6w_1 + w_2 - 0.6\}, \quad (2.16a)$$

$$W_{12} \geq \max\{0.6w_1, 2w_1 + w_2 - 2\}, \quad (2.16b)$$

$$T_{12} \leq \min\{1.6w_1, -t_2\}, \quad (2.16c)$$

$$T_{12} \geq \max\{0, 1.6w_1 - t_2 - 1.6\}, \quad (2.16d)$$

$$W_{11} \leq w_1, \quad (2.16e)$$

$$W_{11} \geq \max\{0, 2w_1 - 1\}, \quad (2.16f)$$

$$W_{22} \leq 2.6w_2 - 1.6t_2 - 1.2, \quad (2.16g)$$

$$W_{22} \geq \max\{1.2w_2 - 0.36, 4w_2 - 4\} + \max\{-3.2t_2 - 2.56, 0\}. \quad (2.16h)$$

These RLT inequalities admit the solution $w_1 = w_2 = 1, t_1 = t_2 = 0, T_{12} = 0, W_{12} = 1, W_{11} = 1, W_{22} = 1.4$, which satisfies constraints (2.12b)-(2.12d) and the positive semidefinite constraint. However, this solution violates valid inequality (2.6a), since $-6W_{11} - W_{22} + 3W_{12} + 1.5T_{12} + 4 = -0.4 < 0$.

Branching on a Complex Matrix Entry

In this section we consider branching on upper and lower bounds of Y as specified in \mathcal{J}_C , i.e., bounds on every complex matrix entry. We examine branching rules for selecting a single (i, j) entry of Y to branch on. One way to form a branching rule is to use a scoring function, where the branching option with the highest score is selected. A score can be based on the violation of relaxed constraints, or an estimate of the impact of branching on the children nodes' optimal relaxation objective value. In the standard development of the SDP relaxation the only relaxed constraint is the rank-one constraint on Y . However, the rank function is discrete and applies globally to all variables of the decision matrix, which seems problematic for use in variable branching. Therefore, we consider an alternative to the rank-one condition:

Proposition 2.2.4. *For $n > 1$ a nonzero Hermitian positive semidefinite $n \times n$ matrix Y has rank one iff all of its 2×2 principal minors are zero.*

Proof. Suppose Y has rank $r > 1$. Since Y is Hermitian it has an $r \times r$ nonzero principal minor. Since Y is positive semidefinite this principal minor corresponds to a positive definite $r \times r$ submatrix. As $r \geq 2$, this implies there exists a 2×2 strictly positive principal minor. Now suppose instead that Y has a strictly positive

2×2 principal minor. Then Y contains a rank-two principal submatrix and thus $r > 1$. \square

We will use the equivalent condition that the minimum eigenvalue of each 2×2 principal submatrix be zero. Algebraically this can be expressed as

$$\lambda_{\min} = \frac{1}{2} \left(W_{ii} + W_{jj} - \|(W_{ii} - W_{jj}, 2W_{ij}, 2T_{ij})\| \right).$$

We branch by partitioning the range $[L_{ij}, U_{ij}]$ for some (i, j) via updating the bounds as:

$$\begin{aligned} L'_{ij} &\leftarrow \alpha L_{ij} + (1 - \alpha)U_{ij} \\ U'_{ij} &\leftarrow \alpha L_{ij} + (1 - \alpha)U_{ij} \end{aligned}$$

where $\alpha \in (0, 1)$ is a parameter. In our implementation we use the bisection rule, i.e., set $\alpha = 0.5$. We will refer to the assignment $L'_{ij} \leftarrow \frac{L_{ij} + U_{ij}}{2}$ as the up branch with respect to a matrix entry, and similarly down branch refers to the assignment on the upper bound. Now let us consider rules for selecting an entry to branch on.

Most Violated with Strong Branching (MVSb)

Let c be a pair of indices (i, j) . Select the 2×2 principal submatrix of Y with the greatest minimum eigenvalue, and let c^* be the column indices of the submatrix. From Proposition 2.2.4 it follows that, for CSDP, $\text{rank}(Y) \in \{0, 1\}$ iff $\lambda_{\min}(Y_c) = 0$ for all pairs. Given some c^* , there are three possible candidate entries. These are evaluated by strong branching: for each entry we will solve the up branch problem and obtain the solution matrix Y^+ , and likewise Y^- from the down branch. The following score function is used:

$$\mu \max\{-\lambda_{\min}(Y_{c^*}^+), -\lambda_{\min}(Y_{c^*}^-)\} + (1 - \mu) \min\{-\lambda_{\min}(Y_{c^*}^+), -\lambda_{\min}(Y_{c^*}^-)\},$$

where $\mu \in [0, 1]$ is a tuning parameter; we follow the example of COUENNE [19] and use a value of 0.15. The entry with the highest score is selected for branching.

Most Violated with Worst-Case Bounds (MVWB)

Since strong branching is computationally expensive, we consider solving a simpler subproblem to produce a score. Consider the Worst-Case Eigenvalue (WEV) problem of finding the greatest minimum eigenvalue that can be obtained within $\text{conv}(\mathcal{J}_C)$:

$$\text{(WEV)} \quad \max \lambda$$

$$\text{s.t.: } \|(W_{11} - W_{22}, 2W_{12}, 2T_{12})\| \leq W_{11} + W_{22} - 2\lambda, \quad (2.5a) - (2.5c), (2.6a) - (2.6b).$$

Note that we dropped the positive semidefinite condition on X since the objective maximizes the minimum eigenvalue. We solve MEV in lieu of solving the children problems Y^+ and Y^- . Thus, overestimates of $\lambda_{\min}(Y_{c^*}^+)$ and $\lambda_{\min}(Y_{c^*}^-)$ are used in the score function. MVWB is otherwise the same as MVSB.

Reliability Branching with Entry Bounds (RBEB)

Since MVSB and MVWB rely on a particular violation metric, for benchmarking purposes we consider a method that is agnostic to the measure of violation. RBEB is an application of the *rb-int-br* rule of Belotti et al. [19].

Reliability branching uses pseudocosts, which capture information about previous branching decisions. Let Φ^+ and Φ^- be matrices containing pseudocosts, estimating the improvement in objective value by branching up or down, respectively. If, at search tree node k , L_{ij}^k is selected for branching up and the objective improves by δ^k in the child node's relaxation, then let $D^k := \delta^k / (\frac{U_{ij}^k - L_{ij}^k}{2} - L_{ij}^k)$ be the per-unit improvement. Φ^+ is the running average of all D^k for up branches, and Φ^- is the running average for down branches.

For a given candidate (i, j) , the following score function is used:

$$(\mu \max\{\Phi^+, \Phi^-\} + (1 - \mu) \min\{\Phi^+, \Phi^-\}) \left(\frac{U_{ij} - L_{ij}}{2} \right).$$

As with MVSB and MVWB, we use a value of $\mu = 0.15$. The candidate with the highest score is selected for branching. Note that we can restrict the set of candidate entries to those with violation, i.e. corresponding to members of 2×2 principal submatrices with strictly positive minimum eigenvalue.

In reliability branching, strong branching is used in lieu of pseudocosts until η evaluations have been performed on a given up or down branch; η is called the reliability parameter [2]. We test for $\eta = 1$, termed RBEB1, and $\eta = 4$, termed RBEB4.

2.3 Computational Experiments

In this section we present the results of experiments on solving ACOPF and BoxQP problems using the spacial branch-and-cut approach described in the previous sections. All experiments herein are conducted with a 2.26 dual-core Intel i3-350M processor and 4 GB main memory. Algorithms are implemented using MATLAB [107]

with model processing performed by YALMIP [102]. Conic programs are solved with MOSEK version 7.1 [9]. IPOPT version 3.11.1 [161] is used as a local solver to obtain primal feasible solutions to QCQP at each search tree node.

All SBC configurations are implemented with a depth-first search node selection rule. The search termination criteria are: an explored nodes limit of 10000, a time limit of 1.5 hours, and a relative optimality gap limit. A search tree depth of 100 is applied, pruning all children nodes past this limit. The optimality gap is calculated using the global upper bound (`gub`) and global lower bound (`glb`) : $\text{gap} = 1 - (\text{gub} - \text{glb})/|\text{gub}|$.

Problem Instances

ACOPF

The ACOPF problem is a power generation scheduling problem that can be formulated as QCQP. The matrix entry bounds specified in \mathcal{J}_c are provided in the form of squared voltage magnitude bounds for diagonal terms and voltage phase angle bounds for off-diagonal terms. The problem formulation will be detailed in Chapter 3.

Our experiments include the test cases of Gopalakrishnan et al. [61]. Small duality gaps were reported for these cases, so the root relaxation is known to provide a good lower bound. These instances are named `g9`, `g14`, `g30`, and `g57`, where the number indicates the number of buses in the problem. As in Gopalakrishnan et al. [61] we set a optimality gap limit of 0.1%. We also use the modified IEEE test cases that will be detailed in Chapter 3, which are named `9Na`, `9Nb`, `14S`, `14P`, and `118IN`. For these more difficult instances we set a optimality gap limit of 1%. Since the IEEE test cases do not include phase angle differences, we have applied a 30 degree bound across all connected buses. We also use a sparse formulation of CSDP that replaces the positive semidefinite constraint with multiple positive semidefinite constraints on submatrices and linear equality constraints. This is described in detail in Chapter 4, which also proves that enforcing the rank constraint on each submatrix is sufficient to ensure equivalence between QCQP and the sparse version of CSDP.

BoxQP

The BoxQP problem is formulated as $\min \frac{1}{2}x'Qx + f'x : 0 \leq x \leq 1$, where $x \in \mathbb{R}^n$ is decision vector, and $f \in \mathbb{R}^n, Q \in \mathbb{R}^{n \times n}$ are data. We use the BoxQP instances of Vandembussche and Nemhauser [158]. The instances are named *sparAAA-BBB-C*,

where AAA is the dimension of x , BBB is the density of Q , and C is the random seed number. We set an optimality gap limit of 0.01% for these instances.

Results

In Table 2.1 we compare the branch selection rules on the Gopalakrishnan instances of ACOPF, which have small root gap when using the CSDP relaxation. The columns are labelled as follows: **case** is the case or instance name, **nodes** are the number of search tree nodes explored before termination, **depth** is the maximum search tree depth, **time** is the total time spent in seconds by the SBC algorithm. The root gap (**rgap**) is calculated as $\text{rgap} = (\text{gub} - \text{rlb}) / |\text{gub}|$, where **gub** is the best known upper bound and **rlb** is the root node lower bound. Since the same relaxation is used for all selection strategies, **rgap** is given a separate column. The end gap (**egap**), is calculated as $\text{egap} = (\text{gub} - \text{glb}) / |\text{gub}|$, where **glb** is the global lower bound established by the SBC algorithm at termination. An asterisk indicates failure to achieve the optimality termination criterion due to an explored nodes, a maximum depth, or a total time limit. The table shows that, despite the strong initial bounds provided by CSDP, practical convergence cannot be achieved using matrix entry bound branching. Note that we exclude MVWB from comparison, as this selection rule uses valid inequalities that are not present in the basic relaxation CSDP. None of the branching rules converge within the limits of the computations.

In Table 2.2 we present the results with experiments with adding the valid inequalities describing $\text{conv}(\mathcal{J}_C)$ to strengthen the standard SDP relaxation. Comparison with Table 2.1 clearly shows the positive impact in the solution times. With the employment of valid inequalities, the algorithm converges much faster and we are able to solve the difficult cases with large root gaps. MVWB performs the best on average. Compared to MVWB, MVSb results in a smaller search tree. The reliability branching rules do not perform as well as the violation-based rules and fail to converge on all difficult instances. Although not shown in the table, we note that adding valid inequalities has negligible effect on the relaxation solution times, with roughly $\pm 5\%$ change in average solution time per instance. Although the root gaps are not improved with the valid inequalities, the inequalities become effective as variable bounds tighten deeper in the search tree. We note that Kocuk, Dey and Sun [86] prove that a standard application of RLT inequalities will not strengthen the standard SDP relaxation of ACOPF.

In Table 2.3 we compare three relaxations for BoxQP: SDP with RLT inequalities on all entries (SDP+RLT); SDP with valid inequalities describing $\text{conv}(\mathcal{J}_R)$ (SDP+VI); and the standard SDP relaxation. The columns are labelled as follows: **case** is the case or instance name, **gap** is the optimality gap between the relaxation

and the optimal objective value. The SDP relaxation solves substantially faster without valid inequalities, but SDP+RLT and SDP+VI produce much tighter bounds. The SDP+RLT has comparable solution times to SDP+VI and closes 85% of the optimality gap of SDP+VI. This is an expected behaviour, as noted in Section 2.2.

In Table 2.4 we compare different branching strategies using the SDP+RLT relaxation; note that we leave out instances that are solved with zero root gap. On average, MVSB uses fewer nodes and less depth to terminate than MVWB, but MVWB results in faster solution times. This is a typical outcome for strong branching. Both MVWB and MVSB are substantially faster than the reliability-based rules RBEB1 and RBEB4. RBEB4 did not reach the optimality criterion for the last case due to a high number of strong branching candidates.

Since the SDP+RLT relaxation produces rather tight bounds, we also solve the BoxQP instances with SDP+VI to test branch selection strategies when the search tree is larger. The results when a weaker relaxation is used are shown in Table 2.5. Similar conclusions can be drawn as with Table 2.4. MVWB and MVSB achieve comparable performance, with a roughly equal trade off between time and search tree size. Both strategies result in better average performance compared to the reliability-based rules.

Table 2.1: Comparison of branching rules using CSDP for ACOFP

case	rgap	MVSB				RBEB1				RBEB4			
		nodes	depth	time	egap	nodes	depth	time	egap	nodes	depth	time	egap
g9	0.36%	10077*	101*	4654	0.36%	6481	101*	5421*	0.36%	1639	101*	5509*	0.36%
g14	0.16%	8235	101*	5427*	0.16%	10041*	101*	5021	0.16%	10035*	101*	5047	0.16%
g30	0.18%	4247	101*	5440*	0.18%	5821	101*	5425*	0.18%	5433	101*	5416*	0.18%
g57	2.31%	379	101*	6348*	2.31%	1765	101*	5475*	2.31%	1005	101*	5553*	2.31%
Average	0.75%	5735	101	5467	0.75%	6027	101	5335	0.75%	4528	101	5381	0.75%

Table 2.2: Comparison of branching rules using CSDP+VI for ACOFP

case	rgap	MVWB				MVSB				RBEB1				RBEB4			
		nodes	depth	time	egap	nodes	depth	time	egap	nodes	depth	time	egap	nodes	depth	time	egap
g9	0.36%	77	27	28	0.10%	65	24	28	0.10%	9935	101*	5423	0.36%	967	61	237	0.10%
g14	0.16%	67	21	28	0.10%	41	14	26	0.09%	165	40	65	0.09%	211	42	80	0.10%
g30	0.18%	75	12	51	0.09%	51	9	66	0.10%	133	55	84	0.10%	175	52	135	0.10%
g57	2.31%	215	68	1202	0.10%	69	26	416	0.09%	865	101*	5588	2.21%	323	57	2630	0.10%
9Na	18.00%	2771	70	587	1.00%	10011*	41	4223	2.16%	10033*	85	2372	15.05%	10037*	99	2651	14.23%
9Nb	19.29%	2325	69	522	1.00%	9435	41	3573	1.00%	10043*	85	3242	14.66%	10029*	71	2416	14.77%
14P	5.32%	4999	72	1766	1.00%	7087	41	5405*	5.20%	8715	101*	5434	5.30%	10041*	101*	2779	5.08%
14S	2.97%	3979	72	1628	1.00%	6397	39	5406*	2.97%	5437	101*	5437	2.96%	10045*	101*	3497	2.90%
118IN	2.05%	241	39	1821	0.99%	265	39	5515*	1.81%	401	101*	5989	2.33%	33	17	5595	1.47%
Average	5.63%	1639	50	848	0.60%	3713	30	2740	1.50%	5081	86	3737	4.79%	4651	67	2224	4.32%

Table 2.3: Comparison of relaxations on BoxQP instances

case	SDP+RLT		SDP+VI		SDP	
	gap	time	gap	time	gap	time
spar020-100-2	0.16%	0.2	0.60%	0.2	91.82%	0.2
spar030-060-1	1.23%	0.5	5.22%	0.6	257.08%	1.2
spar030-060-3	0.36%	0.5	2.48%	0.5	123.62%	0.4
spar030-070-1	3.06%	0.7	18.92%	0.6	322.56%	3.1
spar030-070-3	0.01%	0.6	0.64%	0.5	69.77%	0.0
spar030-080-1	1.31%	0.9	8.88%	0.6	258.73%	1.3
spar030-100-2	0.05%	0.6	0.82%	0.6	143.54%	0.1
spar030-100-3	0.13%	0.6	2.23%	0.6	156.83%	0.1
spar040-040-1	3.12%	2.3	7.44%	1.9	300.53%	3.1
spar040-040-3	0.63%	2.2	1.90%	1.8	165.06%	0.6
spar040-050-1	0.51%	2.1	6.58%	1.9	312.95%	0.5
spar040-050-2	0.35%	2.5	1.25%	2.2	186.91%	0.4
spar040-060-1	2.29%	1.8	6.21%	2.1	246.54%	2.3
spar040-080-3	0.01%	2.1	0.53%	1.8	105.83%	0.0
spar040-090-2	0.03%	2.2	0.36%	1.6	156.48%	0.0
spar040-100-2	0.18%	2	2.29%	2	176.90%	0.2
spar040-100-3	2.26%	2	9.33%	2.7	222.10%	2.3
spar050-030-2	0.20%	6.1	1.02%	6	153.73%	0.2
spar050-030-3	0.08%	5.1	1.58%	5.1	180.61%	0.1
spar050-040-2	0.21%	5	1.66%	5.6	161.57%	0.2
spar050-050-1	8.66%	4.6	35.42%	5	400.90%	8.7
spar050-050-2	0.76%	6.4	3.57%	4.9	208.22%	0.8
spar050-050-3	0.75%	4.5	1.92%	6.2	176.47%	0.8
spar060-020-3	0.54%	11.6	1.30%	13.3	176.59%	0.5
Average	1.12%	2.8	5.09%	2.8	197.68%	1.1

Table 2.4: Comparison of branching rules with SDP+RLT relaxation on BoxQP

case	rgap	MVWB				MVSB				RBEB1				RBEB4			
		nodes	depth	time	egap	nodes	depth	time	egap	nodes	depth	time	egap	nodes	depth	time	egap
spar020-100-2	0.16%	3	2	1	0.00%	3	2	1	0.00%	29	15	13	0.00%	39	20	33	0.00%
spar030-060-1	1.21%	5	3	3	0.00%	3	2	6	0.00%	15	8	43	0.00%	17	9	152	0.00%
spar030-060-3	0.36%	15	8	10	0.00%	5	3	8	0.00%	39	20	63	0.00%	41	21	179	0.00%
spar030-070-1	2.97%	39	8	28	0.00%	19	6	40	0.00%	29	12	56	0.00%	27	13	164	0.00%
spar030-080-1	1.30%	7	3	5	0.00%	3	2	6	0.00%	11	6	39	0.00%	17	9	148	0.00%
spar030-100-2	0.05%	3	2	2	0.00%	1	1	3	0.00%	7	4	37	0.01%	9	5	140	0.00%
spar030-100-3	0.13%	3	2	2	0.00%	3	2	4	0.00%	53	27	69	0.00%	67	34	187	0.00%
spar040-040-1	3.02%	31	7	68	0.00%	13	5	91	0.00%	59	24	285	0.00%	51	16	787	0.00%
spar040-040-3	0.62%	5	3	11	0.00%	3	2	14	0.00%	9	5	176	0.00%	9	5	648	0.00%
spar040-050-1	0.51%	5	3	11	0.00%	3	2	14	0.00%	37	19	235	0.00%	39	20	705	0.00%
spar040-050-2	0.35%	3	2	7	0.00%	1	1	11	0.00%	53	27	283	0.00%	59	30	776	0.00%
spar040-060-1	2.24%	23	5	46	0.00%	11	4	89	0.00%	45	14	236	0.00%	53	18	693	0.00%
spar040-080-3	0.01%	3	2	6	0.00%	3	2	11	0.00%	29	15	199	0.00%	29	15	613	0.00%
spar040-090-2	0.03%	3	2	6	0.00%	1	1	9	0.00%	31	16	224	0.00%	31	16	671	0.00%
spar040-100-2	0.18%	5	3	10	0.00%	3	2	13	0.00%	21	11	195	0.00%	21	11	648	0.00%
spar040-100-3	2.21%	27	6	53	0.01%	25	6	181	0.00%	31	13	203	0.00%	33	11	676	0.00%
spar050-030-2	0.20%	5	3	29	0.00%	5	3	68	0.00%	45	23	789	0.00%	51	26	2626	0.00%
spar050-030-3	0.08%	3	2	15	0.00%	1	1	25	0.00%	11	6	580	0.01%	13	7	2328	0.01%
spar050-040-2	0.21%	5	3	27	0.00%	1	1	26	0.00%	15	8	591	0.01%	17	9	2186	0.01%
spar050-050-1	7.97%	439	14	2339	0.01%	189	11	3874	0.00%	365	18	2409	0.00%	287	16	3544	0.00%
spar050-050-2	0.76%	9	4	48	0.00%	7	3	141	0.00%	21	11	624	0.01%	21	11	2140	0.01%
spar050-050-3	0.75%	13	5	68	0.00%	5	3	88	0.00%	39	20	720	0.00%	41	21	2213	0.00%
spar060-020-3	0.54%	11	4	134	0.00%	3	2	93	0.00%	15	8	1695	0.00%	9	5	6190*	0.30%
Average	1.08%	28	4	122	0.00%	13	3	201	0.00%	43	14	407	0.00%	42	15	1186	0.01%

Table 2.5: Comparison of branching rules with SDP+VI relaxation on BoxQP

case	rgap	MVWB				MVSB				RBEB1				RBEB4			
		nodes	depth	time	egap	nodes	depth	time	egap	nodes	depth	time	egap	nodes	depth	time	egap
spar020-100-2	2.17%	69	13	18	0.00%	37	8	23	0.00%	49	17	19	0.01%	51	15	35	0.00%
spar030-060-1	3.30%	39	11	27	0.00%	9	4	23	0.00%	65	21	78	0.00%	59	24	174	0.01%
spar030-060-3	2.58%	247	23	170	0.01%	57	11	137	0.01%	345	71	294	0.01%	343	77	392	0.01%
spar030-070-1	5.12%	257	21	195	0.00%	53	13	134	0.00%	295	73	268	0.00%	309	75	375	0.00%
spar030-080-1	2.71%	21	6	15	0.00%	11	4	23	0.00%	59	23	75	0.01%	43	17	159	0.00%
spar030-100-1	0.70%	21	10	15	0.01%	7	3	16	0.01%	57	29	71	0.01%	59	30	166	0.01%
spar030-100-2	2.17%	17	7	12	0.00%	7	3	13	0.00%	19	9	45	0.00%	19	9	135	0.00%
spar040-040-1	5.61%	455	24	973	0.01%	189	17	1410	0.01%	683	101*	1616	0.02%	643	97	1978	0.01%
spar040-040-2	0.04%	5	3	11	0.00%	1	1	10	0.01%	33	17	235	0.01%	33	17	738	0.01%
spar040-050-1	2.25%	33	9	70	0.01%	19	7	121	0.00%	109	43	388	0.01%	109	46	827	0.01%
spar040-050-2	1.64%	59	9	129	0.01%	23	5	163	0.01%	135	52	452	0.00%	145	56	919	0.01%
spar040-060-1	4.61%	343	17	741	0.01%	135	11	953	0.01%	501	76	1237	0.01%	701	74	2153	0.01%
spar040-080-1	0.95%	31	16	67	0.00%	5	3	36	0.00%	51	26	262	0.01%	49	25	750	0.01%
spar040-090-2	1.49%	25	8	54	0.00%	9	4	67	0.00%	61	26	278	0.01%	87	23	866	0.00%
spar040-100-2	2.00%	71	19	154	0.01%	27	10	191	0.01%	169	53	509	0.01%	187	53	992	0.01%
spar040-100-3	4.04%	527	26	1166	0.01%	79	10	634	0.01%	829	56	1887	0.00%	713	52	2125	0.01%
spar050-030-2	1.88%	143	18	853	0.01%	37	8	799	0.01%	167	56	1550	0.00%	159	59	3135	0.00%
spar050-030-3	1.93%	211	18	1198	0.00%	39	9	850	0.01%	163	53	1481	0.00%	153	47	3036	0.00%
spar050-040-2	2.00%	71	11	393	0.01%	37	8	849	0.01%	197	44	1710	0.01%	119	47	2874	0.00%
spar050-050-1	9.97%	1047	18	5423*	5.64%	285	15	5449*	6.71%	987	33	5433*	2.88%	697	28	5433*	7.68%
spar050-050-2	3.12%	147	24	850	0.00%	59	11	1248	0.01%	223	57	1786	0.01%	195	52	3215	0.01%
spar050-050-3	2.70%	417	19	2407	0.01%	133	10	2416	0.01%	615	57	4192	0.01%	545	58	5425	0.01%
spar060-020-3	2.74%	413	21	5479*	1.31%	121	12	5532*	1.11%	293	56	5605*	1.13%	9	5	7006*	1.99%
Average	2.78%	195	15	851	0.29%	58	8	879	0.33%	257	45	1229	0.17%	229	42	1789	0.41%

2.4 Conclusion

This chapter presented a spatial branch-and-cut approach for generic Quadratically-Constrained Quadratic Programs with bounded complex variables. We derived valid inequalities from the convex hull description of nonconvex rank-one restricted sets to strengthen the SDP relaxations. We gave a new branching method based on an alternative characterization of a rank-one constraint. Experiments on Alternating Current Optimal Power Flow problems showed that the valid inequalities are critical to improving the performance of the algorithm. The proposed branching methods resulted in better performance compared to the benchmark reliability branching method. Tests on box-constrained nonconvex Quadratic Programming instances suggest that the violation-based branching methods may also be effective for problems with real variables.

Chapter 3

Bound Tightening for the Alternating Current Optimal Power Flow Problem

3.1 Introduction

In this chapter we will improve the spatial branch-and-cut algorithm of Chapter 2 by adding bound-tightening procedures. We focus on a single-period scheduling problem that incorporates steady-state AC power, the Alternating Current Optimal Power Flow (ACOPF). A standard ACOPF problem is to find a minimum cost dispatch of generation and transmission assets to supply load, subject to engineering constraints. Since the AC power flow equations are nonlinear, a common approach to solving ACOPF is through iterative Newton-type solvers (e.g. [153]), which can only guarantee local optimality. Linearization approaches (see [8]) suffer from the same problems. Therefore, in terms of global optimality, the performance of ACOPF methods used in practice remain an open question. The feasible set of ACOPF is nonconvex (see [69]), and it is NP-hard (see [159, 92]). Even finding a feasible solution to ACOPF for radial instances with fixed voltages is NP-hard (see [94]).

An important challenge in power systems optimization is to accurately model electricity in alternating current (AC) networks while maintaining computational tractability. Multi-period problems employ coarser approximations, where transmission losses are eschewed and power flows are linearized so that efficient algorithms can be employed. For certain purposes linearization can perform adequately (see [124, 151, 96]). However, the inaccuracies that stem from these approximations can result in suboptimal and even infeasible solutions, which may be unacceptable in

other cases. It is unclear how much room for improvement may be made by better accounting for AC power flow. However, as Mixed-Integer Programming software for Unit Commitment has shown, even small improvements in operations can have significant overall impact (see [123]).

Interest in conic optimization techniques for ACOPF is largely due to multiple reports of zero duality gap for various IEEE power system test cases (see [130, 92, 16]). Zero duality gap means that the optimal value of a problem instance coincides with that of the corresponding dual. Throughout the chapter we will be referring to the Lagrangian dual of the ACOPF problem formulated as a nonconvex Quadratically-Constrained Quadratic Program (QCQP). The Lagrangian dual of any QCQP can be solved with semidefinite programming (SDP). SDP is a type of conic optimization problem, which is a useful paradigm for global optimization as it guarantees that any local optimum is also globally optimal. Conic optimization can be done with robust methods that can automatically find initial starting points, and have polynomial-time convergence towards the global optimal solution. Even when duality gap exists for ACOPF, by property of weak duality conic relaxations provide a lower bound on the global optimal value, which can be used to judge the quality of candidate feasible solutions. Thus, unlike ACOPF algorithms used in practice, conic optimization can prove a problem is infeasible, or prove that a solution is a global optimum.

For certain network topologies, simple formulations ACOPF can be solved exactly using a conic relaxation (see [92, 169, 149]). However, numerous examples demonstrate that zero duality gap cannot be guaranteed in general (see [95, 31, 85]). In cases with positive duality gap, more advanced techniques must be developed to search for global optimal solutions, such as higher moment relaxations (see [75, 112]). The method of higher moment relaxations involves solving a sequence of convex problems that grow rapidly in problem size. Several papers have considered an alternative method to global optimization, a spatial branch-and-bound algorithm (see [130, 61, 85]). These algorithms all use lifted relaxations that are computationally expensive to solve for large instances. Therefore practical global optimization for ACOPF remains an active research area.

This chapter provides computationally efficient methods for bound tightening, namely the tightening of voltage magnitude, line flow, and phase angle limits. Bound tightening reduces the domain of a problem by removing infeasible regions. This can improve the quality of a relaxation while maintaining its validity; this is true of any relaxation, whether SDP, SOCP, or LP-based (e.g. [42, 27]). For computational experiments we have applied bound tightening to the spatial branch-and-cut algorithm introduced in Chapter 2. Although domain reduction has a natural application to global optimization as it can tighten relaxations, it can be applied elsewhere. For instance, bound tightening could improve the warm-start for an iterative optimiza-

tion method, or it may complement the low-rank SDP-based heuristic introduced in Sojoudi, Madani and Lavaei [150]. In addition to the bound tightening methods, we also construct modified IEEE test cases with large duality gap that may pose a challenge to global optimization algorithms. Computational tests using a complex QCQP solver (see [39]) show that bound tightening improves the solver's convergence rate on these hard problems. The instances are made publicly available at <https://sites.google.com/site/cchenresearch/>.

The rest of the chapter is organized as follows: Section 3.2 describes ACOPF and a SDP relaxation for it; Section 3.3 introduces new ACOPF instances with large duality gap; Section 3.4 details the bound tightening procedures; Section 3.5 contains computational results; Section 3.6 concludes the chapter.

3.2 Formulations

We present a basic optimal power flow formulation:

$$(\mathbf{ACOPF}) : \min c_2^T [P + D_P]^2 + c_1^T (P + D_P) + c_0$$

subject to

$$P + jQ = \text{diag}(Y^* V V^*), \quad (3.1a)$$

$$P^{\min} \leq P + D_P \leq P^{\max}, \quad (3.1b)$$

$$Q^{\min} \leq Q + D_Q \leq Q^{\max}, \quad (3.1c)$$

$$[V^{\min}]^2 \leq \text{diag}(V V^*) \leq [V^{\max}]^2, \quad (3.1d)$$

$$\tan(\theta_{mn}^{\min}) \text{Re}(V_m V_n^*) \leq \text{Im}(V_m V_n^*) \leq \tan(\theta_{mn}^{\max}) \text{Re}(V_m V_n^*). \quad (3.1e)$$

Let n be the number of nodes in the graph of the problem, with nodes representing either buses or transformers, and let k be the number of edges (aka branches). The decision variables used are: nodal powers $P + jQ \in \mathbb{C}^n$; a Hermitian decision matrix representing the outer product of nodal voltages, $W + iT \in \mathbb{H}^{n \times n}$; and power to and from buses (respectively) across branches, $P_f + jQ_f, P_t + jQ_t \in \mathbb{C}^k$. All other parameters are fixed data: convex costs $c_0 \in \mathbb{R}, c_1 \in \mathbb{R}^N, c_2 \in \mathbb{R}_+^N$; load, $D_P + jD_Q \in \mathbb{C}^n$; admittance matrices, $Y \in \mathbb{C}^{n \times n}, Y_f, Y_t \in \mathbb{C}^{k \times n}$; voltage magnitude limits, $V^{\min}, V^{\max} \in \mathbb{R}^n$; phase angle limits $\theta^{\min}, \theta^{\max} \in \mathbb{R}^k$; generator limits, $P^{\min} + jQ^{\min}, P^{\max} + jQ^{\max} \in \mathbb{C}^n$; and line limits, $S^{\max} \in \mathbb{R}_{++}^n$. $Y := G + jB$ is the bus admittance matrix, and $Y_f := G_f + jB_f, Y_t := G_t + jB_t$ are branch admittances corresponding to 'from' and 'to' nodes, respectively. Admittance is composed of

conductance G and susceptance B . For a branch r from m to n , the (r, m) entry of $C_f \in \mathbb{R}^{k \times n}$ and the (r, n) entry of $C_t \in \mathbb{R}^{k \times n}$ are 1; all unconnected entries in those matrices are 0 [172].

The objective is to minimize the cost of real power generation, where $P + P^D$ is the net generation of real nodal power. The power flow equations are modeled with constraint (3.1a), and nodal and generation power limits with constraints (3.1b) and (3.1c). Constraint (3.1d) enforces voltage magnitude limits, and constraint (3.1e) enforces bus angle difference limits. Bus angle differences can be recovered with $\theta_{mn} = \arctan(\text{Im}(V_m V_n^*) / \text{Re}(V_m V_n^*))$. Note that for notational brevity we have left out line limits, but three types are explicitly considered in Section 3.4.

Following Lavaei and Low [92], we consider the following lifted SDP relaxation:

$$(\mathbf{RACOPF}) : \min c_2^T [P + D_P]^2 + c_1^T (P + D_P) + c_0$$

subject to

$$P = \text{diag}(GW - BT), \quad (3.2a)$$

$$Q = \text{diag}(-BW - GT), \quad (3.2b)$$

$$P^{\min} \leq P + D_P \leq P^{\max}, \quad (3.2c)$$

$$Q^{\min} \leq Q + D_Q \leq P^{\max}, \quad (3.2d)$$

$$[V^{\min}]^2 \leq \text{diag}(W) \leq [V^{\max}]^2, \quad (3.2e)$$

$$\tan(\theta_{mn}^{\min})W_{mn} \leq T_{mn} \leq \tan(\theta_{mn}^{\max})W_{mn}, \quad (3.2f)$$

$$W + iT \succeq 0. \quad (3.2g)$$

The decision vector V has been replaced by the Hermitian decision matrix $W + iT$, and a rank-one condition on $W + iT$ has been relaxed.

3.3 New Instances With Large Duality Gap

First we provide some intuition regarding the construction of cases with duality gap. Provided ACOPF is feasible, RACOPF has the same optimal cost if and only if there exists an optimal solution to RACOPF with rank 1. Recall the alternative condition from Chapter 2:

Proposition 3.3.1. *For $n > 1$ a nonzero Hermitian positive semidefinite $n \times n$ matrix X has rank one iff all its 2×2 principal minors are zero.*

Suppose we are given an ACOPF-optimal solution with voltages \hat{V} and powers \hat{P}, \hat{Q} . Consider its corresponding rank-one optimal solution in lifted space $\hat{W} + j\hat{T} = \hat{V}\hat{V}^*$. From the proposition we have that the principal minor condition $\hat{W}_{mn}^2 + \hat{T}_{mn}^2 = \hat{W}_{mm}\hat{W}_{nn}$ holds across all bus pairs. The positive semidefinite constraint (3.2g) enforces $\hat{W}_{mn}^2 + \hat{T}_{mn}^2 \leq \hat{W}_{mm}\hat{W}_{nn}$, so a gap between LACOPF and ACOPF can only occur if for each optimal solution to LACOPF there exists at least one pair m, n such that $W_{mn}^2 + T_{mn}^2 < W_{mm}W_{nn}$.

If there is a pair m, n where decreasing either \hat{T}_{mn} or \hat{W}_{mn} improves the objective value, then there is a gap between the optimal values of LACOPF and ACOPF. A decrease in the magnitude of \hat{T}_{mn} has the equivalent effect of a decrease in the magnitude of θ_{mn} . Note that this is a nonphysical effect if the rank condition is lost; for instance this could lead to an increase in power factor between m and n without affecting flows elsewhere. Decreasing \hat{W}_{mn} decreases real and reactive power at both buses by decreasing real and reactive power flows in both directions across the connecting lines.

Let us now consider conditions where decreasing the magnitudes of \hat{T}_{mn} or \hat{W}_{mn} (and adjusting nodal powers accordingly) could improve the objective function. Since decreasing $|\hat{T}_{mn}|$ reduces the power transfer between m and n , then it may be desirable when line congestion is problematic, or when transfer across a lossy transmission line is otherwise unavoidable. Decreasing \hat{W}_{mn} increases losses, which would allow an otherwise unmanageable amount of power to be produced by dissipating flows in an unphysical manner. Using this intuition, we construct new cases with large duality gap by applying simple changes to IEEE test cases. We name these cases as follows: *9Na*, *9Nb*, *14S*, *14P*, and *118IN*, with number indicating the number of buses and letter indicating the type of change.

Negative costs: *9Na* and *9Nb*

We use the 9-bus instance in MATPOWER and change the cost coefficients by setting all quadratic coefficients c_2 to be zero, and reversing the sign of the linear real power cost coefficients c_1 on certain generators, making these costs negative. Negative cost coefficients can model opportunity costs such as start-up and shut-down cost avoidance, ramping considerations (e.g. anticipating high demand in the next period), feed-in-tariffs from renewable resources, and the value of absorbing excess generation from an import bus. Thus the cost coefficients in these cases represent bids rather than marginal generation costs. For this small 9-bus network we have constructed extreme cases: in *9NA* we give negative costs to generators 1 and 3, and in *9NB* we do so for all three generators.

Congestion: 14S and 14P

We modified the IEEE 14-bus case by applying a universal line limit across all lines, applying either real power (14P) or apparent power (14S) limits. For 14P we apply a per-unit limit of 0.23, and for 14S a per-unit limit of 0.24. These produce severe amounts of congestion, as further lowering the limit on either case by 0.01 resulted in infeasibility.

Congestion and Negative Costs: 118IN

We modified the 118-bus IEEE case in order to construct a relatively large case with modest duality gap. The first (in lexicographic order) 7 generators were set to have negative linear costs, and all quadratic cost coefficients were set to zero. Substantial congestion was created by setting a thermal limit across all lines of 1.14 p.u. on current magnitude.

3.4 Bound Tightening Procedures for ACOPF

In this section we propose fast procedures for domain reduction aka bound tightening. The typical procedure for bound tightening is as follows: minimize/maximize the desired variable subject to the constraints of the relaxation; however, this is computationally intensive. Instead we consider ACOPF-specific methods with closed-form solutions.

Tightening on power flows

Let us examine some bus m . If we consider voltage magnitude and angle constraints at all buses, and real and reactive power constraints only at bus m , then we have the following ACOPF relaxation in polar coordinates:

$$V_n^{\min} \leq |V_n| \leq V_n^{\max}, \quad (3.3a)$$

$$\theta_{mn}^{\min} \leq \theta_{mn} \leq \theta_{mn}^{\max}, \quad (3.3b)$$

$$\begin{aligned} P_m^{\min} &\leq G_{mm}|V_m|^2 + \sum_{\forall n \in \mathcal{C}(m)} G_{mn}|V_m||V_n| \cos(\theta_{mn}) \\ &+ \sum_{\forall n \in \mathcal{C}(m)} B_{mn}|V_m||V_n| \sin(\theta_{mn}) \leq P_m^{\max}, \end{aligned} \quad (3.3c)$$

$$\begin{aligned} Q_m^{\min} &\leq -B_{mm}|V_m|^2 - \sum_{\forall n \in \mathcal{C}(m)} B_{mn}|V_m||V_n| \cos(\theta_{mn}) \\ &+ \sum_{\forall n \in \mathcal{C}(m)} G_{mn}|V_m||V_n| \sin(\theta_{mn}) \leq Q_m^{\max}. \end{aligned} \quad (3.3d)$$

We can further relax the problem by decoupling P and Q, and rewriting some terms using the optimal solution to certain subproblems. First let us define the following terms:

$$\begin{aligned} p_{mn} &:= G_{mn}|V_n| \cos(\theta_{mn}) + B_{mn}|V_n| \sin(\theta_{mn}), \\ q_{mn} &:= -B_{mn}|V_n| \cos(\theta_{mn}) + G_{mn}|V_n| \sin(\theta_{mn}), \\ p_m &:= \sum_{\forall n \in \mathcal{C}(m)} p_{mn}, \\ q_m &:= \sum_{\forall n \in \mathcal{C}(m)} q_{mn}. \end{aligned}$$

Thus we can rewrite the nodal power equations:

$$\begin{aligned} P_m &= G_{mm}|V_m|^2 + p_m|V_m|, \\ Q_m &= -B_{mm}|V_m|^2 + q_m|V_m|. \end{aligned}$$

We can obtain upper and lower bounds on p_{mn}, q_{mn} by finding the following

optima:

$$\begin{aligned}
 p_{mn}^U &:= \max G_{mn}|V_n| \cos(\theta_{mn}) + B_{mn}|V_n| \sin(\theta_{mn}) \\
 &\text{subject to constraints (3.3a) and (3.3b),} \\
 p_{mn}^L &:= \min G_{mn}|V_n| \cos(\theta_{mn}) + B_{mn}|V_n| \sin(\theta_{mn}) \\
 &\text{subject to constraints (3.3a) and (3.3b),} \\
 q_{mn}^U &:= \max -B_{mn}|V_n| \cos(\theta_{mn}) + G_{mn}|V_n| \sin(\theta_{mn}) \\
 &\text{subject to constraints (3.3a) and (3.3b),} \\
 q_{mn}^L &:= \min -B_{mn}|V_n| \cos(\theta_{mn}) + G_{mn}|V_n| \sin(\theta_{mn}) \\
 &\text{subject to constraints (3.3a) and (3.3b),} \\
 p_m^L &:= \sum_{\forall n \in \mathcal{C}(m)} p_{mn}^L, \\
 p_m^U &:= \sum_{\forall n \in \mathcal{C}(m)} p_{mn}^U, \\
 q_m^L &:= \sum_{\forall n \in \mathcal{C}(m)} q_{mn}^L, \\
 q_m^U &:= \sum_{\forall n \in \mathcal{C}(m)} q_{mn}^U.
 \end{aligned}$$

Each bound is computed by checking variable bounds and the unconstrained first-order necessary conditions (FONC). For instance, for p_{mn} , FONC give us either $\theta_{mn}^* = \arctan(\frac{B_{mn}}{G_{mn}})$ if $G_{mn} \neq 0$, or else $\theta_{mn}^* = \frac{\pi}{2}$, in which case we can discard the point as infeasible. We can test all candidates ($\theta_{mn} = \arctan(\frac{B_{mn}}{G_{mn}}), |V_n| = |V_n|^{\min}, \theta_{mn} = \theta_{mn}^{\max}, |V_n| = |V_n|^{\max}$, etc.) to determine bounds for p_m, q_m . With these variable bounds we form the following relaxation:

$$\begin{aligned}
 V_m^{\min} &\leq |V_m| \leq V_m^{\max}, \\
 P_m^{\min} &\leq G_{mm}|V_m|^2 + p_m|V_m| \leq P_m^{\max}, \\
 Q_m^{\min} &\leq -B_{mm}|V_m|^2 + q_m|V_m| \leq Q_m^{\max}, \\
 p_m^L &\leq p_m \leq p_m^U, \\
 q_m^L &\leq q_m \leq q_m^U.
 \end{aligned}$$

From here, new variable bounds are determined in a straightforward and computationally efficient manner. Let us consider the power bounds first. An (resp. over)

underestimate of the (resp. max) min (resp. real) reactive power flow is attained either at variable bounds or the appropriate unconstrained FONC point. For P we have:

$$\begin{aligned}
 \min\{P^N, P^A\} &\leq G_{mm}|V_m|^2 + p_m|V_m| \leq \max\{P^X, P^B\}, \\
 P^N &:= \min\{G_{mm}(V_m^{\min})^2 + p_m^L(V_m^{\min}), \\
 &G_{mm}(V_m^{\max})^2 + p_m^L(V_m^{\max})\}, \\
 P^X &:= \max\{G_{mm}(V_m^{\min})^2 + p_m^U(V_m^{\min}), \\
 &G_{mm}(V_m^{\max})^2 + p_m^U(V_m^{\max})\}, \\
 V_P^A &:= \begin{cases} -\frac{p_m^L}{2G_{mm}}, & G_{mm} \neq 0, V_m^{\min} \leq -\frac{p_m^L}{2G_{mm}} \leq V_m^{\max}, \\ V_m^{\min} & o/w, \end{cases} \\
 V_P^B &:= \begin{cases} -\frac{p_m^U}{2G_{mm}}, & G_{mm} \neq 0, V_m^{\min} \leq -\frac{p_m^U}{2G_{mm}} \leq V_m^{\max}, \\ V_m^{\max} & o/w, \end{cases} \\
 P^A &:= G_{mm}(V_P^A)^2 + p_m^L(V_P^A), \\
 P^B &:= G_{mm}(V_P^B)^2 + p_m^U(V_P^B).
 \end{aligned}$$

Here, V_P^A, V_P^B are FONC solutions for V_m given a fixed value of p_m , and P^A, P^B are the corresponding nodal powers. Reactive power can be updated in the same way, replacing G_{mm} with $-B_{mm}$, and p with q . Thus voltage limits can be used to tighten nodal power limits.

We can apply the quadratic root formula to make inferences about voltage magnitude using real and reactive power constraints. Note that only the positive root needs to be considered, as the negative part corresponds to the lower portion of the nose curve that is avoided in power systems operation to maintain stability. Let us consider the following problem structure:

$$\begin{aligned}
 ax^2 + bx + c &\leq 0, \\
 b^L &\leq b \leq b^U, \\
 &\equiv \\
 ax^2 + bx + c + d &= 0, \\
 d &\geq 0, \\
 b^L &\leq b \leq b^U.
 \end{aligned}$$

Here, a, c are parameters, and x, b, d are real-valued variables. For example, with the real power upper bound we have $a = G_{mm}, b = p_m, c = -P^{\max}, x = |V_m|$, and hence tightening of the limits on voltage magnitude. We are interested in the projection on x so that we can establish implied variable bounds for purposes of tightening: $x^L \leq x \leq x^U$. We use the property that the lower portion of nose curves are forbidden or infeasible regions and take only the higher root:

$$x = \frac{\sqrt{b^2 - 4ac - 4ad}}{2|a|} - \frac{b}{2a}.$$

From here, by maximizing or minimizing the right-hand side with b, d , we infer either upper and lower bounds x^L, x^U , or else infeasibility:

$$a = 0, b^U < 0 :$$

$$-\frac{c}{b^L} \leq x,$$

$$a = 0, b^L > 0, c \leq 0 :$$

$$x \leq -\frac{c}{b^L},$$

$$a = 0, b^L > 0, c > 0 :$$

Infeasibility,

$$a < 0 :$$

$$-\frac{\min\{\sqrt{[(b^L)^2 - 4ac]^+} + b^L, \sqrt{[(b^U)^2 - 4ac]^+} + b^U\}}{2a} \leq x,$$

$$\begin{aligned}
 & a > 0, 4ac \leq (b^N)^2 : \\
 & -\frac{b^U}{2a} \leq x \leq \frac{\max\{\sqrt{(b^L)^2 - 4ac} - b^L, \sqrt{(b^U)^2 - 4ac} - b^U\}}{2a}, \\
 & a > 0, (b^N)^2 < 4ac \leq (b^X)^2, b^X < 0 : \\
 & \sqrt{\frac{c}{a}} \leq x \leq \frac{\sqrt{(b^L)^2 - 4ac} - b^L}{2a}, \\
 & a > 0, (b^N)^2 < 4ac \leq (b^X)^2, b^X > 0 : \\
 & -\frac{b^U}{2a} \leq x \leq \frac{\sqrt{(b^U)^2 - 4ac} - b^U}{2a}, \\
 & a > 0, 4ac > (b^X)^2 : \\
 & \text{Infeasibility,} \\
 & b^N := \begin{cases} b^L, & (b^L)^2 < (b^U)^2, \\ b^U & \text{o/w,} \end{cases} \\
 & b^X := \begin{cases} b^L, & (b^L)^2 > (b^U)^2, \\ b^U & \text{o/w.} \end{cases}
 \end{aligned}$$

Tightening on line constraints

Three types of line flow limits typically used for ACOPF are apparent power, real power, and line current magnitude. For notational simplicity we will assume that nodes are connected by a single line. This is not a technical requirement (we include limits on specific lines in our experiments) as multiple lines can be easily accommodated by replacing m, n entries with the appropriate indices for specific lines.

Apparent power is usually applied as a proxy for thermal line or transformer limits: $P_{mn}^2 + Q_{mn}^2 \leq (S_{mn}^{\max})^2$

Note that line limits are quartic constraints with respect to voltages. In our relaxation we include nodal powers as explicit decision variables, so the line limits are modeled as convex quadratic constraints with respect to power, which maintains the QCQP framework.

Let us now deduce some limits by fixing either P_{mn} or Q_{mn} in the same way as with nodal powers. For brevity we will only show the procedure for real power, with the procedure holding symmetrically for reactive power. First let us determine the minimum possible magnitude of reactive power flow:

$$\begin{aligned}
 Q_{mn}^0 &:= \max\{Q_{mn}^N, \min\{Q_{mn}^X, 0\}\}, \\
 Q_{mn}^N &:= \min\{-B_{mn}(V_m^{\min})^2 - q_{mn}^U V_m^{\min}, \\
 &\quad -B_{mn}(V_m^{\max})^2 - q_{mn}^U V_m^{\max}, \\
 &\quad -B_{mn}(V_Q^A)^2 - q_{mn}^U V_Q^A\}, \\
 Q_{mn}^X &:= \max\{-B_{mn}(V_m^{\min})^2 - q_{mn}^L V_m^{\min}, \\
 &\quad -B_{mn}(V_m^{\max})^2 - q_{mn}^L V_m^{\max}, \\
 &\quad -B_{mn}(V_Q^B)^2 - q_{mn}^L V_Q^B\}, \\
 V_Q^A &:= \begin{cases} -\frac{q_{mn}^U}{2B_{mn}}, & B_{mn} \neq 0, V_m^{\min} \leq -\frac{q_{mn}^U}{2B_{mn}} \leq V_m^{\max}, \\ V_m^{\min} & o/w, \end{cases} \\
 V_Q^B &:= \begin{cases} -\frac{q_{mn}^L}{2B_{mn}}, & B_{mn} \neq 0, V_m^{\min} \leq -\frac{q_{mn}^L}{2B_{mn}} \leq V_m^{\max}, \\ V_m^{\max} & o/w. \end{cases}
 \end{aligned}$$

Using the same principles as with nodal powers, we have determined Q_{mn}^N, Q_{mn}^X , respectively upper and lower bounds on reactive power flow from m to n . $|Q_{mn}^0|$ gives us the minimum magnitude, so we have the following valid inequality:

$$\begin{aligned}
 P_{mn}^2 + (Q_{mn}^0)^2 &\leq (S_{mn}^{\max})^2, \\
 &\equiv P_{mn}^{\min} \leq P_{mn} \leq P_{mn}^{\max}, \\
 P_{mn}^{\min} &:= -\sqrt{(S_{mn}^{\max})^2 - (Q_{mn}^0)^2}, \\
 P_{mn}^{\max} &:= \sqrt{(S_{mn}^{\max})^2 - (Q_{mn}^0)^2}.
 \end{aligned}$$

Note that the explicit real power limits are sometimes included as a proxy for voltage stability limits. In such a case we use whichever bound is tighter. Voltage magnitude bound tightening can then be applied to line flow limits using the same procedure as for nodal power limits. For instance, with P_{mn}^{\max} and we have $G_{mn}|V_m|^2 - p_{mn}|V_m| - P_{mn}^{\max} \leq 0$.

Line current magnitude is the key factor in thermal line limit violation. In the

lifted space we write this as a linear constraint, unlike apparent power bounds:

$$\begin{aligned}
 |I_{mn}| &\leq I_{mn}^{\max}, \\
 &\equiv (G_{mn}^2 + B_{mn}^2)(|V_m|^2 + |V_n|^2 \\
 &\quad - 2|V_m||V_n|\cos(\theta_{mn})) \leq (I_{mn}^{\max})^2, \\
 &\implies (G_{mn}^2 + B_{mn}^2)(W_{mm} + W_{nn}, \\
 &\quad - 2W_{mn}) \leq (I_{mn}^{\max})^2.
 \end{aligned}$$

Defining $\theta_{mn}^0 := \max\{\theta_{mn}^{\min}, \min\{\theta_{mn}^{\max}, 0\}\}$ as the feasible angle closest to zero, we make the following inference, supposing that $|Y_{mn}| \neq 0$:

$$\begin{aligned}
 |V_m| &\leq \max\{V^D, V^E, V^F\}, \\
 V^C &:= \begin{cases} \sqrt{\frac{I^R}{1-\cos^2(\theta_{mn}^0)}}, & \theta_{mn}^0 \neq 0, \\ \infty, & o/w, \end{cases} \\
 V^D &:= \begin{cases} \frac{V_n^{\max} \cos(\theta_{mn}^0)}{+\sqrt{I^R + (V_n^{\max})^2(\cos^2(\theta_{mn}^0) - 1)}}, & V_n^{\max} \leq V^C, \\ V^C, & o/w, \end{cases} \\
 V^E &:= \begin{cases} \frac{V_n^{\min} \cos(\theta_{mn}^0)}{+\sqrt{I^R + (V_n^{\min})^2(\cos^2(\theta_{mn}^0) - 1)}}, & V_n^{\min} \leq V^C, \\ \text{Infeasible problem}, & o/w, \end{cases} \\
 V^F &:= \min\{V_n^{\min}, \max\{\min\{V_n^{\max}, V^C\}, V^G\}\}, \\
 V^G &:= \begin{cases} \frac{\sqrt{I^R}(|\sin(\theta_{mn}^0)| + \cos(\theta_{mn}^0)|\cot(\theta_{mn}^0)|)}{V^E}, & \theta_{mn}^0 \neq 0, \\ V^E, & o/w, \end{cases} \\
 I^R &:= \frac{(I_{mn}^{\max})^2}{G_{mn}^2 + B_{mn}^2}.
 \end{aligned}$$

The same procedure applies symmetrically on $|V_n|$. We also update the angle bounds, considering only the nontrivial cases where $|Y_{mn}|V_m^{\min}V_n^{\min} \neq 0$:

$$\begin{aligned}
 &(G_{mn}^2 + B_{mn}^2)(|V_m|^2 + |V_n|^2, \\
 &\quad - 2|V_m||V_n|\cos(\theta_{mn})) \leq (I_{mn}^{\max})^2, \\
 &\implies \frac{|V_m|^2 + |V_n|^2 - I^R}{2|V_m||V_n|} \leq \cos(\theta_{mn}). \tag{3.4}
 \end{aligned}$$

We can then determine the minimum of the left-hand side of inequality (3.4). Supposing a nontrivial bound, where $(V_m^{\min})^2 + (V_n^{\min})^2 - I^R > 0$, $V_m^{\min} > 0$, $V_n^{\min} > 0$, by examining derivatives we can see that the minimum is attained at one of the following values:

$$\begin{aligned} |V_m| = V_m^{\min}, |V_n| &= \max\{V_n^{\min}, \min\{V_n^{\max}, (V_m^{\min})^2 - I^R\}\}, \\ |V_m| = V_m^{\max}, |V_n| &= \max\{V_n^{\min}, \min\{V_n^{\max}, (V_m^{\max})^2 - I^R\}\}, \\ |V_n| = V_n^{\min}, |V_m| &= \max\{V_m^{\min}, \min\{V_m^{\max}, (V_n^{\min})^2 - I^R\}\}, \\ |V_n| = V_n^{\max}, |V_m| &= \max\{V_m^{\min}, \min\{V_m^{\max}, (V_n^{\max})^2 - I^R\}\}. \end{aligned}$$

Let θ_{mn}^A be the minimum value for the left-hand side of inequality (3.4). If $\theta_{mn}^A > 1$, then the problem is infeasible; otherwise, θ_{mn} is bounded above and below by $\pm \arccos(\theta_{mn}^A)$.

Tightening on graph cycles

For meshed networks, we propagate angle bound changes across cycles using the identity that the sum of angle differences around a cycle must sum to zero. For instance, if we choose to partition the bounds of θ_{mn} , it is easy to check if there is some third bus k that connects to m and n in the chordally completed graph. Supposing that all bounds are at -30 to 30 degrees, and that the upper bound $\hat{\theta}_{mn}^{\max}$ has been updated to -15 degrees, then we can update the other lower bounds:

$$\begin{aligned} \hat{\theta}_{nk}^{\min} &= \max\{\theta_{nk}^{\min}, -(\hat{\theta}_{mn}^{\max} + \theta_{km}^{\max})\} \\ &= -15 \text{ deg}, \end{aligned}$$

$$\begin{aligned} \hat{\theta}_{km}^{\min} &= \max\{\theta_{km}^{\min}, -(\hat{\theta}_{mn}^{\max} + \theta_{nk}^{\max})\} \\ &= -15 \text{ deg}. \end{aligned}$$

Although this applies to cycles of any size, for simplicity we restrict the procedure to 3-cycles in our experiments. Note that this procedure generalizes to QCQP with bounded complex variables (see [39]).

3.5 Computational Experiments

Setup

All experiments herein are performed with a 2.26 dual-core Intel i3-350M processor and 4 GB main memory. Algorithms are coded in MATLAB [107] with model processing performed by YALMIP [102]. We use the solver for QCQP with bounded complex variables developed in Chapter 2.

In addition to the challenging instances produced in Section 3.3, we have included test instances with small duality gap from Gopalakrishnan et al. [61]. These instances are called *g9-g57*, with the number indicating the number of buses in the problem. Since the IEEE test cases do not include phase angle difference limits, we have applied a 30 degree bound for all connected bus pairs. For the challenging problems we set a global optimality tolerance of 1%, and for *g9-g57* we use a tolerance of 0.1%.

Results

We summarize our results using MVWB in Table 3.1. The columns are defined as follows. *Case* is the case name. *BT* indicates whether the bound tightening procedures were used. *Nodes* are the number of search tree nodes explored before termination. *Depth* is the maximum search tree depth. *Time* is the total time spent in the solver. *rgap* is the root gap, calculated as $(gub - rlb)/|gub|$, where *gub* is the best known upper bound and *rlb* is the root node lower bound. *egap* is the end gap, calculated as $(gub - glb)/|gub|$, where *glb* is the global lower bound established by the solver at termination. *cgap* is the closed gap, i.e., $1 - \frac{egap}{rgap}$. Note that *egap-rgap* is a lower bound for the duality gap present in these cases. Column averages are provided in the last row. For cases with small root gap, *g9-g30*, bound tightening has a modest effect, and the solver terminates quickly regardless. For the more difficult problems bound tightening presents clear advantages in both time and search tree size. In case *118IN* bound tightening reduces the root gap by 16%, and for other cases it does not have substantial effect at the root node.

Table 3.2 provides a detailed breakdown of times. *Total* indicates the total time spent in the solver. *LB* is the time spent solving lower bound problems, *UB* is the time spent upper bound problems. The overhead (*OH*) is calculated as $Total - LB - UB$. $Total/n$ is the time spent per search tree node, and OH/n is the overhead time per node. There are small but significant increases in overhead due to the bound tightening procedures. The per-node overhead increase is larger on the more difficult cases. This is because the bound tightening procedure is able to prune a high percentage of nodes due to infeasibility, and thus makes up a larger percentage of total

Table 3.1: Comparison with and without bound tightening

case	With Tightening						Without Tightening					
	nodes	depth	time	rgap	egap	cgap	nodes	depth	time	rgap	egap	cgap
g9	57	22	20	0.36%	0.08%	76%	77	27	28	0.36%	0.10%	73%
g14	49	18	18	0.16%	0.10%	38%	67	21	28	0.16%	0.10%	40%
g30	73	12	45	0.18%	0.09%	49%	75	12	51	0.18%	0.09%	49%
g57	109	39	358	2.31%	0.09%	96%	215	68	1202	2.31%	0.10%	96%
9Na	1171	45	340	18.00%	1.00%	94%	2771	70	587	18.00%	1.00%	94%
9Nb	1149	43	333	19.29%	1.00%	95%	2325	69	522	19.29%	1.00%	95%
14P	2290	50	1026	5.32%	1.00%	81%	4999	72	1766	5.32%	1.00%	81%
14S	1799	51	968	2.97%	1.00%	66%	3979	72	1628	2.97%	1.00%	66%
118IN	190	33	1620	1.61%	1.00%	38%	241	39	1821	2.05%	0.99%	52%
Average	699	37	925	4.56%	0.71%	65%	1608	51	1561	4.59%	0.79%	60%

time spent. With the exception of phase angle tightening the procedures presented in this chapter involve closed-form solutions and involves only simple arithmetic operations. The angle tightening involves propagation across cycles on chordal graphs (see [39]), which has linear time worst case complexity (see [29]). Thus it may be possible to further reduce per-node time impacts with more efficient coding practices than used in our prototype.

Table 3.3 contains the results of applying bound tightening procedures for all branching rules. Again, modest improvements can be seen for the cases with small root gap. For the more difficult cases, bound tightening offers substantial improvements, with both MVWB and MVSB able to reach the optimality criterion for all cases. The observed increase in overhead as a result of the bound tightening procedures is less than 5% of total time. Again, reliability branching rules do not perform as well as the violation strategies. We also note that bound tightening reduces the root gap by 21% for 118IN.

3.6 Conclusion

We constructed new instances of ACOPF with high duality gap, which require methods beyond a one-stage SDP relaxation for global optimization. We presented closed-form bound-tightening procedures to reduce the domain of the problem by removing infeasible regions. Computational experiments using a spatial branch-and-cut solver indicate that the bound-tightening techniques are particularly effective on more difficult instances. Further experiments could explore details regarding configurations, e.g. number of bound tightening passes per node, size of cycles for angle tightening, and applying strong tightening at the root node.

Table 3.2: Breakdown of time spent (seconds)

case	With Tightening						Without Tightening					
	Total	LB	UB	OH	Total/n	OH/n	Total	LB	UB	OH	Total/n	OH/n
g9	20	3	11	6	0.35	0.11	28	5	16	7	0.36	0.09
g14	18	5	8	4	0.37	0.09	28	9	14	5	0.42	0.08
g30	45	18	17	10	0.62	0.14	51	20	22	9	0.68	0.12
g57	358	114	190	54	3.28	0.49	1202	233	911	59	5.59	0.27
9Na	340	58	173	109	0.29	0.09	587	134	307	147	0.21	0.05
9Nb	333	57	168	107	0.29	0.09	522	117	274	131	0.22	0.06
14P	1026	316	420	290	0.45	0.13	1766	648	811	307	0.35	0.06
14S	968	263	436	269	0.54	0.15	1628	548	804	275	0.41	0.07
118IN	1620	737	553	330	8.52	1.74	1821	878	638	304	7.56	1.26
Average	925	241	497	187	4.07	0.60	1561	472	885	204	2.80	0.29

Table 3.3: Comparison of branching rules using CSDP+VI and bound tightening for ACOPF

case	rgap	MVWB				MVSB				RBEB1				RBEB4			
		nodes	depth	time	egap	nodes	depth	time	egap	nodes	depth	time	egap	nodes	depth	time	egap
g9	0.36%	57	22	20	0.08%	52	21	28	0.10%	7545	82	5440*	0.36%	607	49	207	0.10%
g14	0.16%	49	18	18	0.10%	31	11	20	0.09%	151	36	97	0.10%	177	36	87	0.10%
g30	0.18%	73	12	45	0.09%	49	9	68	0.10%	125	54	141	0.10%	248	52	241	0.10%
g57	2.31%	109	39	358	0.09%	51	23	386	0.10%	784	101*	5531*	2.22%	264	88	2456	0.10%
9Na	18.00%	1171	45	340	1.00%	460	30	195	0.99%	10030*	73	3461	14.72%	7188	61	1780	1.00%
9Nb	19.29%	1149	43	333	1.00%	722	29	313	1.00%	10052*	101*	3555	14.56%	6464	58	1690	1.00%
14P	5.32%	2290	50	1026	1.00%	1582	31	1454	1.00%	10048*	101*	4897	5.31%	10033*	97	3499	5.08%
14S	2.97%	1799	51	968	1.00%	1187	28	1307	1.00%	10041*	101*	4879	2.96%	10024*	95	4473	2.63%
118IN	1.61%	190	33	1620	1.00%	250	21	5980	1.00%	562	91	5614*	1.46%	264	81	5670*	1.41%
Average	5.58%	765	35	525	0.60%	487	23	1083	0.60%	5482	82	3735	4.64%	3919	69	2234	1.28%

Chapter 4

Sparse Cuts for a Positive Semidefinite Constraint

In this chapter we study valid inequalities for a positive semidefinite constraint on a Hermitian decision matrix. This is the canonical constraint of complex semidefinite programming (SDP), which as mentioned in Chapter 2 can be used to solve a relaxation of complex quadratically-constrained quadratic programs. The valid inequalities can be used to form an outer-approximation of any complex or real semidefinite program. We will apply these inequalities on the Alternating Current Optimal Power Flow (ACOPF) problem, which has sparse structure that can be incorporated into its SDP relaxation [113, 72, 15]. We propose a separation approach to solving a sparse complex (or real) SDP problem, which allows for the flexibility of terminating at a near-optimal solution and retaining a weaker but potentially easier to solve relaxation. This may be particularly useful for branch-and-cut methods, although may also be benefits to incorporating cuts into interior point methods (see [111, 88]). Specific applications include large-scale problems, mixed-integer SDP, and spatial branch-and-bound where a SDP relaxation may be solved repeatedly.

The rest of the chapter is organized as follows: Section 4.1 describes the standard linear cuts for SDP, Section 4.2 details the proposed conic cuts, Section 4.3 describes the sparse decomposition approach, Section 4.4 provides some computations, Section 4.5 concludes the chapter.

4.1 Linear Valid Inequalities

A Hermitian matrix $X \in \mathbb{C}^{n \times n}$ has only real eigenvalues, which we will order as follows: $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n$. Assign a corresponding eigenbasis: d_1, d_2, \dots, d_n . We can

use the Courant-Fischer min-max principle to form a variational characterization of these eigenvalues (see [70]):

$$\lambda_i(X) = \max_{\dim(c)=i} \min_{c \in \mathcal{C}: \|c\|=1} c^* X c$$

Consequently, since X is positive semidefinite only when its minimum eigenvalue is nonnegative, the positive semidefinite constraint is equivalent to a semi-infinite linear constraint set:

$$X \succeq 0 \equiv c^* X c \geq 0 \quad \forall c \in \mathcal{C}$$

Thus, if some optimal decision matrix \hat{X} is not PSD, then we can introduce cutting planes $c^* X c \geq 0, c \in \hat{\mathcal{C}}$, where $\hat{\mathcal{C}}$ is some (finite) set of valid inequalities. Motivated by the min-max principle, Sherali and Tuncbilek [145] proposed using all eigenvectors corresponding to the negative eigenvalues of \hat{X} for real SDP problems; this approach was later extended to sparse SDP by Qualizza, Belotti, and Margot [133]. This is a method of exact separation, as cuts can be generated iff \hat{X} is not positive semidefinite. Hence any SDP may be outer-approximated to arbitrary precision using these cuts. This method adds the deepest cut in some sense, since from the variational characterization we have that d_n attains maximal violation:

$$\lambda_n(X) = d_n^* X d_n = \min_{c \in \mathbb{R}^n: \|c\|=1} c^* X c$$

We note that a small modification is used to apply these cuts — originally proposed for real-valued symmetric matrices — to Hermitian matrices. One can use two real $n \times n$ matrices of decision variables for the real and imaginary components, $X := \text{Re}(X) + i\text{Im}(X)$, where the real part is symmetric, and the imaginary skew-symmetric. Any cut with c can be implemented as a linear constraint using real decision variables: $\text{Re}(c)^* \text{Re}(X) \text{Re}(c) + \text{Im}(c)^* \text{Re}(X) \text{Im}(c) + 2\text{Im}(c)^* \text{Im}(X) \text{Re}(c) \geq 0$.

4.2 Second-Order Cone Valid Inequalities

Another necessary condition for a matrix to be positive semidefinite is:

$$X \succeq 0 \equiv N^* X N \succeq 0 \implies C^* X C \succeq 0.$$

N a $n \times n$ nonsingular matrix and C some matrix with n rows. Oskoorouchi and Mitchell [122] observed that if C is $n \times 2$ then $C^T X C$ is a 2×2 matrix, and so this positive semidefinite constraint can be written as a second-order cone constraint. However, they left open the question of separating from this set, i.e. selecting a C . We present some analysis that can guide cut generation.

Let us define the cut coefficient C in terms of its columns: $C = [c_1 \ c_2], C \in \mathbb{C}^{n \times 2}$. Thus

$$C^* X C = \begin{bmatrix} c_1^* X c_1 & c_1^* X c_2 \\ c_2^* X c_1 & c_2^* X c_2 \end{bmatrix}$$

$C^*XC \succeq 0$ is equivalent to the following three inequalities:

$$(c_1^*Xc_1)(c_2^*Xc_2) \geq |c_1^*Xc_2|^2 \quad (4.1a)$$

$$c_1^*Xc_1 \geq 0, c_2^*Xc_2 \geq 0 \quad (4.1b)$$

Note that inequality 4.1a ensures nonnegativity of $\det(C^*XC)$, and inequalities 4.1b ensure nonnegativity of the remaining principal minors. These describe a rotated second-order cone, and so they are equivalent to the following second-order cone constraint:

$$c_1^*Xc_1 + c_2^*Xc_2 \geq \|(2c_1^*Xc_2, c_1^*Xc_1 - c_2^*Xc_2)\| \quad (4.2)$$

Inequality 4.2 constraints the minimum eigenvalue of X since $\lambda_{\min}(C^*XC) = \frac{1}{2}(c_1^*Xc_1 + c_2^*Xc_2 - \|(2c_1^*Xc_2, c_1^*Xc_1 - c_2^*Xc_2)\|)$. Using the transformation described in the case of linear valid inequalities, this cut can be implemented using real decision variables. It has several desirable properties. Since inequality 4.2 implies 4.1b, then the second-order cone cut dominates the linear cut. Furthermore, this approach subsumes the 2×2 principal minor valid inequalities that are commonly used to form a SOCP relaxation of SDP (e.g. [80]):

$$X \succeq 0 \implies \begin{bmatrix} X_{pp} & X_{pq} \\ X_{qp} & X_{qq} \end{bmatrix} \succeq 0 \forall p, q.$$

For the principal minor corresponding to indices p, q , one can set $c_1 = e_p, c_2 = e_q$, where e_p is the unit basis vector with an entry of 1 at p and 0 everywhere else. Hence there is a possibility of generating a stronger SOCP relaxation for a given SDP. To address this question we examine the violation of a given cut. Let $\|\cdot\|_F$ denote the Frobenius norm. Given a symmetric matrix $\hat{X} \in \mathbb{R}^{n \times n}$ with eigenvalues $\lambda_1 \geq \dots \geq \lambda_n$, and some $C \in \mathbb{R}^{n \times 2}$, we have:

Theorem 4.2.1.

1. $\min_{\|C\|_F=1} \lambda_{\min}(C^*\hat{X}C) = \lambda_n$
2. $\min_{\|C\|_F=1} \det(C^*\hat{X}C) = \begin{cases} \frac{\lambda_1\lambda_n}{4}, & \lambda_1 > 0 \\ 0, & \lambda_1 \leq 0 \end{cases}$

Proof. We start with the first minimization. Let $C = [c_1, c_2]$, and let d_1, d_2, \dots, d_n be an eigenbasis of X , where each d_i corresponds to the eigenvectors of X , $\lambda_1 \geq \dots \geq \lambda_n$.

Then with a change of variables $a, b \in \mathbb{R}^n : a_i \triangleq c_1^* d_i, b_i \triangleq c_2^* d_i$ we have:

$$\begin{aligned}
 z_1^* &:= \min_{\|C\|_F=1} \lambda_{\min}(C^* X C) \\
 &= \min_{c_1^* c_1 + c_2^* c_2 = 1} \frac{1}{2} [c_1^* X c_1 + c_2^* X c_2 \\
 &\quad - \|(c_1^* X c_1 - c_2^* X c_2, 2c_1^* X c_2)^T\|] \\
 &= \min_{a^T a + b^T b = 1} \frac{1}{2} \left[\sum_{i=1}^n \lambda_i (a_i^2 + b_i^2) \right. \\
 &\quad \left. - \left\| \left(\sum_{i=1}^n \lambda_i (a_i^2 - b_i^2), 2 \sum_{i=1}^n \lambda_i a_i b_i \right)^T \right\| \right] \tag{4.3}
 \end{aligned}$$

First suppose that $\sum_{i=1}^n \lambda_i a_i b_i = 0$. Then

$$\begin{aligned}
 z_1^* &= \min_{a^T a + b^T b = 1} \frac{1}{2} \left[\sum_{i=1}^n \lambda_i (a_i^2 + b_i^2) - \left\| \sum_{i=1}^n \lambda_i (a_i^2 - b_i^2) \right\| \right] \\
 &= \min_{a^T a + b^T b = 1} \frac{1}{2} \left[\sum_{i=1}^n \lambda_i a_i^2 + \sum_{i=1}^n \lambda_i b_i^2 - \left| \sum_{i=1}^n \lambda_i a_i^2 - \sum_{i=1}^n \lambda_i b_i^2 \right| \right] \\
 &= \min_{a^T a + b^T b = 1} \left\{ \sum_{i=1}^n \lambda_i a_i^2, \sum_{i=1}^n \lambda_i b_i^2 \right\} \\
 &= \lambda_n
 \end{aligned}$$

The penultimate equation is obtained using the fact that $x + y - |x - y| = \min\{2x, 2y\}$.

Now suppose $\sum_{i=1}^n \lambda_i a_i b_i \neq 0$ and let μ be the Lagrange multiplier associated with $a^T a + b^T b - 1 = 0$. We have the following first order necessary conditions for optimality for all $k = 1, \dots, n$:

$$\mu a_k = \lambda_k \left[a_k - \frac{a_k \sum_{i=1}^n \lambda_i (a_i^2 - b_i^2) + 2b_k \sum_{i=1}^n \lambda_i a_i b_i}{\|(\sum_{i=1}^n \lambda_i (a_i^2 - b_i^2), 2 \sum_{i=1}^n \lambda_i a_i b_i)^T\|} \right] \tag{4.4}$$

$$\mu b_k = \lambda_k \left[b_k - \frac{2a_k \sum_{i=1}^n \lambda_i a_i b_i - b_k \sum_{i=1}^n \lambda_i (a_i^2 - b_i^2)}{\|(\sum_{i=1}^n \lambda_i (a_i^2 - b_i^2), 2 \sum_{i=1}^n \lambda_i a_i b_i)^T\|} \right] \tag{4.5}$$

By supposition there must exist some $a_m b_m \lambda_m \neq 0$, so we have:

$$\mu = \lambda_m - \lambda_m \frac{\sum_{i=1}^n \lambda_i (a_i^2 - b_i^2) + 2 \frac{b_m}{a_m} \sum_{i=1}^n \lambda_i a_i b_i}{\|(\sum_{i=1}^n \lambda_i (a_i^2 - b_i^2), 2 \sum_{i=1}^n \lambda_i a_i b_i)^T\|} \tag{4.6}$$

$$\mu = \lambda_m - \lambda_m \frac{2 \frac{a_m}{b_m} \sum_{i=1}^n \lambda_i a_i b_i - \sum_{i=1}^n \lambda_i (a_i^2 - b_i^2)}{\|(\sum_{i=1}^n \lambda_i (a_i^2 - b_i^2), 2 \sum_{i=1}^n \lambda_i a_i b_i)^T\|} \tag{4.7}$$

Together, (4.6)-(4.7) imply the following:

$$\sum_{i=1}^n \lambda_i (a_i^2 - b_i^2) = \left(\frac{a_m}{b_m} - \frac{b_m}{a_m} \right) \sum_{i=1}^n \lambda_i a_i b_i \quad (4.8)$$

Substituting (4.8) back into (4.6), we have:

$$\begin{aligned} \mu &= \lambda_m - \lambda_m \frac{\left(\frac{a_m}{b_m} - \frac{b_m}{a_m} \right) \sum_{i=1}^n \lambda_i a_i b_i + 2 \frac{b_m}{a_m} \sum_{i=1}^n \lambda_i a_i b_i}{\left\| \left(\left(\frac{a_m}{b_m} - \frac{b_m}{a_m} \right) \sum_{i=1}^n \lambda_i a_i b_i, 2 \sum_{i=1}^n \lambda_i a_i b_i \right)^T \right\|} \\ &= \lambda_m - \lambda_m \frac{\left(\frac{a_m}{b_m} + \frac{b_m}{a_m} \right) \sum_{i=1}^n \lambda_i a_i b_i}{\left\| \left(\frac{a_m}{b_m} + \frac{b_m}{a_m} \right) \sum_{i=1}^n \lambda_i a_i b_i \right\|} \\ &= \lambda_m - \lambda_m \frac{\left(\frac{a_m}{b_m} + \frac{b_m}{a_m} \right) \sum_{i=1}^n \lambda_i a_i b_i}{\left| \left(\frac{a_m}{b_m} + \frac{b_m}{a_m} \right) \sum_{i=1}^n \lambda_i a_i b_i \right|} \\ &= \begin{cases} 0, & \left(\frac{a_m^2 + b_m^2}{a_m b_m} \right) \sum_{i=1}^n \lambda_i a_i b_i > 0 \\ 2\lambda_m, & \left(\frac{a_m^2 + b_m^2}{a_m b_m} \right) \sum_{i=1}^n \lambda_i a_i b_i < 0 \end{cases} \quad (4.9) \end{aligned}$$

Now multiplying (4.4) by a_k and (4.5) by b_k and taking a sum over all k we obtain:

$$\begin{aligned} \mu &= \mu \left(\sum_{k=1}^n a_k^2 + b_k^2 \right) \\ &= \sum_{k=1}^n \lambda_k (a_k^2 + b_k^2) \\ &\quad - \frac{\sum_{k=1}^n \lambda_k (a_k^2 - b_k^2) \sum_{i=1}^n \lambda_i (a_i^2 - b_i^2)}{\left\| \left(\sum_{i=1}^n \lambda_i (a_i^2 - b_i^2), 2 \sum_{i=1}^n \lambda_i a_i b_i \right)^T \right\|} \\ &\quad - \frac{4 \sum_{k=1}^n \lambda_k a_k b_k \sum_{i=1}^n \lambda_i a_i b_i}{\left\| \left(\sum_{i=1}^n \lambda_i (a_i^2 - b_i^2), 2 \sum_{i=1}^n \lambda_i a_i b_i \right)^T \right\|} \\ &= \sum_{i=1}^n \lambda_i (a_i^2 + b_i^2) - \left\| \left(\sum_{i=1}^n \lambda_i (a_i^2 - b_i^2), 2 \sum_{i=1}^n \lambda_i a_i b_i \right)^T \right\| \quad (4.10) \end{aligned}$$

Thus μ is twice the optimal objective value, and so by (4.10)-(4.9) we have:

$$z_1^* = \begin{cases} 0, & \left(\frac{a_m^2 + b_m^2}{a_m b_m} \right) \sum_{i=1}^n \lambda_i a_i b_i > 0 \\ \lambda_m, & \left(\frac{a_m^2 + b_m^2}{a_m b_m} \right) \sum_{i=1}^n \lambda_i a_i b_i < 0 \end{cases}$$

Hence $z_1^* \geq \lambda_n$, and equality can be attained by setting $a_n = 1$, so $z_1^* = \lambda_n$.

We prove the second minimization using similar arguments:

$$\begin{aligned}
 z_2^* &= \min_{\|C\|_F=1} \det(C^*XC) \\
 &= \min_{c_1^*c_1+c_2^*c_2=1} c_1^*Xc_1c_2^*Xc_2 - (c_1^*Xc_2)^2 \\
 &= \min_{a^T a+b^T b=1} \sum_{i=1}^n \lambda_i a_i^2 \sum_{i=1}^n \lambda_i b_i^2 - \left(\sum_{i=1}^n \lambda_i a_i b_i\right)^2
 \end{aligned} \tag{4.11}$$

First suppose $\sum_{i=1}^n \lambda_i a_i b_i = 0$. Then:

$$\begin{aligned}
 z_2^* &= \min_{a^T a+b^T b=1} \sum_{i=1}^n \lambda_i a_i^2 \sum_{i=1}^n \lambda_i b_i^2 \\
 &= \begin{cases} \frac{\lambda_1 \lambda_n}{4}, & \lambda_1 > 0 \\ 0, & \lambda_1 \leq 0 \end{cases}
 \end{aligned}$$

Now suppose $\sum_{i=1}^n \lambda_i a_i b_i \neq 0$. Reusing μ as the Lagrangian multiplier associated with $a^T a + b^T b - 1 = 0$, we have the following first order necessary conditions for all $k = 1, \dots, n$:

$$\mu a_k = \lambda_k a_k \sum_{i=1}^n \lambda_i b_i^2 - \lambda_k b_k \sum_{i=1}^n \lambda_i a_i b_i \tag{4.12}$$

$$\mu b_k = \lambda_k b_k \sum_{i=1}^n \lambda_i a_i^2 - \lambda_k a_k \sum_{i=1}^n \lambda_i a_i b_i \tag{4.13}$$

Multiplying (4.12) by a_k and (4.13) by b_k , and taking a sum over all k , we have:

$$\begin{aligned}
 \mu &= \mu \sum_{i=1}^n (a_i^2 + b_i^2) \\
 &= \sum_{k=1}^n \lambda_k a_k^2 \sum_{i=1}^n \lambda_i b_i^2 + \sum_{k=1}^n \lambda_k b_k^2 \sum_{i=1}^n \lambda_i a_i^2 \\
 &\quad - 2 \sum_{k=1}^n \lambda_k a_k b_k \sum_{i=1}^n \lambda_i a_i b_i \\
 &= 2 \left(\sum_{i=1}^n \lambda_i a_i^2 \sum_{i=1}^n \lambda_i b_i^2 - \left(\sum_{i=1}^n \lambda_i a_i b_i\right)^2 \right)
 \end{aligned}$$

Thus μ is twice the objective value. If we multiply (4.12) by b_k and (4.13) by a_k , we get:

$$\begin{aligned}
 & \lambda_k a_k b_k \sum_{i=1}^n \lambda_i b_i^2 - \lambda_k b_k^2 \sum_{i=1}^n \lambda_i a_i b_i \\
 &= \lambda_k a_k b_k \sum_{i=1}^n \lambda_i a_i^2 - \lambda_k a_k^2 \sum_{i=1}^n \lambda_i a_i b_i \\
 &\Leftrightarrow \lambda_k (a_k^2 - b_k^2) \sum_{i=1}^n \lambda_i a_i b_i = \lambda_k a_k b_k \sum_{i=1}^n \lambda_i (a_i^2 - b_i^2) \tag{4.14}
 \end{aligned}$$

Suppose $\sum_{i=1}^n \lambda_i (a_i^2 - b_i^2) = 0$, so that together with (4.14) and $\sum_{i=1}^n \lambda_i a_i b_i \neq 0$ we have $a_k^2 = b_k^2 \forall k$. Let $z \in \{-1, 1\}$ be a sign variable so that $a_k = z_k b_k$. Then by (4.11) we have:

$$\begin{aligned}
 z_2^* &= \min_{a^T a + b^T b = 1} \sum_{i=1}^n \lambda_i a_i^2 \sum_{i=1}^n \lambda_i b_i^2 - \left(\sum_{i=1}^n \lambda_i a_i b_i \right)^2 \\
 &= \min_{a^T a + b^T b = 1} \left(\sum_{i=1}^n \lambda_i a_i^2 \right)^2 - \left(\sum_{i=1}^n \lambda_i a_i b_i \right)^2 \\
 &= \min_{a^T a + b^T b = 1} \left(\sum_{i=1}^n \lambda_i a_i^2 + \sum_{i=1}^n \lambda_i a_i b_i \right) \left(\sum_{i=1}^n \lambda_i a_i^2 - \sum_{i=1}^n \lambda_i a_i b_i \right) \\
 &= \min_{a^T a = \frac{1}{2}, z \in \{-1, 1\}} \left(\sum_{i=1}^n 2\lambda_i z_i a_i^2 \right) \left(2 \sum_{i=1}^n \lambda_i (1 - z_i) a_i^2 \right) \tag{4.15}
 \end{aligned}$$

Now if X has no positive eigenvalues, then the optimal value is 0, since each sum in (4.15) must be nonnegative. Otherwise, denoting $p < n$ as the index of the least nonnegative eigenvalue, we have:

$$\begin{aligned}
 z_2^* &= 4 \min_{a^T a = \frac{1}{2}} \left(\sum_{i=1}^p \lambda_i a_i^2 \right) \left(\sum_{j=p+1}^n \lambda_j a_j^2 \right) \\
 &= \frac{\lambda_1 \lambda_n}{4} \tag{4.16}
 \end{aligned}$$

Now suppose $\sum_{i=1}^n \lambda_i (a_i^2 - b_i^2) \neq 0$. Together with the supposition that $\sum_{i=1}^n \lambda_i a_i b_i \neq 0$ and (4.14) we have for some $a_m^2 \neq b_m^2$:

$$\sum_{i=1}^n \lambda_i a_i b_i = \frac{a_m b_m}{a_m^2 - b_m^2} \sum_{i=1}^n \lambda_i (a_i^2 - b_i^2) \tag{4.17}$$

Substituting (4.17) into (4.12) yields:

$$\begin{aligned}\mu a_m &= \lambda_m a_m \sum_{i=1}^n \lambda_i b_i^2 - \lambda_m b_m \frac{a_m b_m}{a_m^2 - b_m^2} \sum_{i=1}^n \lambda_i (a_i^2 - b_i^2) \\ \Leftrightarrow \mu &= \lambda_m \sum_{i=1}^n \lambda_i \left(\frac{a_m^2}{a_m^2 - b_m^2} b_i^2 - \frac{b_m^2}{a_m^2 - b_m^2} a_i^2 \right) \\ &= \frac{\lambda_m}{a_m^2 - b_m^2} \sum_{i \neq m} \lambda_i (a_m^2 b_i^2 - b_m^2 a_i^2)\end{aligned}$$

Since μ is twice the objective value we can write:

$$z_2^* = \min_{a^T a + b^T b = 1} \frac{\lambda_m}{2(a_m^2 - b_m^2)} \sum_{i \neq m} \lambda_i (a_m^2 b_i^2 - b_m^2 a_i^2) \quad (4.18)$$

Assigning again μ as the multiplier for $a^T a + b^T b - 1 = 0$, we have the following first order necessary conditions for all $k \neq m$:

$$\begin{aligned}2\mu a_k &= -\frac{\lambda_m b_m^2 \lambda_k a_k}{a_m^2 - b_m^2} \\ 2\mu b_k &= \frac{\lambda_m a_m^2 \lambda_k b_k}{a_m^2 - b_m^2}\end{aligned}$$

Thus $\lambda_k a_k b_k = 0 \forall k \neq m$. If $\lambda_k a_k = 0$ or $\lambda_k b_k = 0 \forall k \neq m$ then by (4.18) the objective value is 0. Otherwise, if there is some $t \neq m$ such that $\lambda_t a_t \neq 0, b_t = 0$, then by (4.12) we have $\mu = \lambda_k \sum_{i=1}^n \lambda_i b_i^2 \geq \min\{0, \lambda_i \lambda_n\}$. Using (4.13) shows the same bound holds for $\lambda_t b_t \neq 0, a_t = 0$. Thus we have shown in all cases $z_2^* \geq \min\{0, \frac{\lambda_1 \lambda_n}{4}\}$, where 0 can be obtained with $a = b$ and $\frac{\lambda_1 \lambda_n}{4}$ with $a_1 = b_n = \frac{1}{\sqrt{2}}$. \square

The constructive proof of the first statement of Theorem 4.2.1 shows that the cut producing the greatest violation given a bounded Frobenius norm on C can actually be obtained by the largest violation linear cut, i.e. the eigenvalue $d_n(\hat{X})$, using, say $C = [d_n(\hat{X}) \ 0]$. If $\lambda_n < \lambda_{n-1}$, then this cut attains a unique minimum. This result seems lacking, as we are left with simply the original linear cut. For this reason, we proved the second statement. The cut exploiting the largest violation of the nonlinear component, represented in valid inequality 4.1a, is attained with $\sqrt{2}C = [d_1(\hat{X}) \ d_n(\hat{X})]$, provided $\lambda_1 > 0$. Moreover, this cut also implies the largest violation linear cut due to the choice of second column. We will call this cut **soc1**. However, $\sqrt{2}C = [d_{n-1}(\hat{X}) \ d_n(\hat{X})]$ maximizes the largest violation of each of the inequalities 4.1b, provided the columns are independent (otherwise the linear cut is

best). Provided that $\lambda_{n-1} < 0$, such a cut would not result in nonlinear violation from inequality 4.1a. We will call this cut **soc2**. Comparing **soc1** and **soc2** we see there is a tension between finding violation in inequality 4.1a and in inequality 4.1b. Using violation as a guideline, we would expect **soc1** to be more effective in later cutting plane algorithm iterations as the relaxation tightens, due to the increasing condition number of the solution matrix as the minimum eigenvalue approaches 0. We leave as an open question whether there is a better metric for generating second-order cone cuts. For instance, **soc2** does not generate nonlinear violation with respect to the current solution, but as a nonlinear constraint it could still become binding in later iterations.

4.3 Positive Semidefinite Completion

In a sparse SDP, the objective and linear constraints use only a small proportion of the decision matrix variables. Hence the positive semidefinite constraint becomes problematic due to its density, which raises the question of whether it is possible to use fewer variables. It turns out that one can reduce the number of variables by increasing the number of positive semidefinite constraints: Fukuda et al. [58] introduced a methodology that relies on a positive semidefinite completion theorem proved by Grone et al [62]. Variations of this approach have been applied to ACOF [113, 72, 15]. We present one variant below.

Consider some partial Hermitian matrix X , where some entries are specified and others are null. We say X has a positive semidefinite *completion* if there exists a full Hermitian positive semidefinite matrix \bar{X} having the same values in all specified entries of X . Now consider the unweighted undirected graph $G(N, E)$ induced by X : the node set N corresponds to the columns, and there is an edge between distinct nodes, $E_{mn} = 1$, only if the corresponding entry X_{mn} is specified. Furthermore, denote the set of maximal cliques $\mathcal{C}(G)$; a maximal clique $C_r \in \mathcal{C}(G)$ is a clique ($E_{mn} = 1 \forall m, n \in C_r, m \neq n$) not contained by any other clique. Denote the submatrix induced by C_r as $X(C_r)$, having only those rows/columns of X corresponding to the index set C_r . Then the positive semidefinite completion theorem gives a sufficient condition for completion based on chordality. A chordal graph is a graph where the largest chordless cycle has 3 nodes.

Theorem (Grone et al. [62]). *Suppose $G(N, E)$ is chordal. Then X has a positive semidefinite completion iff $X(C_r) \succeq 0 \forall C_r \in \mathcal{C}(G)$.*

While this has wider application, we mention only the one pertinent to our use. Consider a standard form SDP, and construct a matrix U that accounts for variables used in the objective or linear constraints, e.g. $U_{mn} = \sum_{\forall i} |A_{mn}|$ or null if this sum

is zero. Then if the graph induced by U is chordal, we can apply the theorem, and replace $X \succeq 0$ with positive semidefinite constraints of the form $X(C_r) \succeq 0 \forall C_r \in \mathcal{C}$. Although there can, in general, be an exponential number of cliques, for chordal graphs these are linearly bounded and can be found in linear time [29]. If the graph is not chordal, we lose sufficiency of positive semidefinite completion, and so a natural question is: what are the fewest extra entries of U or, equivalently, edges in E needed to ensure G is chordal? Unfortunately [166] proved that this problem, known as minimum triangulation or minimum fill-in, is NP-complete.

However, while a minimum fill-in is desirable, it is not necessary, as any solution (known as a chordal extension) will do, with the complete graph being a trivial case. In this chapter we employ a symbolic Cholesky decomposition on a permutation of U , call it U' . The permutation is found using a popular heuristic called minimum-degree ordering. It turns out that symbolic Cholesky decomposition (Cholesky factorization that is independent of numerical values) will always yield a lower triangular matrix L , where $L + L^T$ that induces a chordal graph. In fact, finding a permutation that results in a Cholesky factorization with the least nonzero entries is equivalent to the minimum triangulation problem. Jabr [72] implemented one a variant of this decomposition technique to greatly reduce solution times to SDP relaxations of ACOF.

As indicated in Kim et al. [81], a similar theorem can be used for LMIs. Consider some symmetric matrix $Y := \sum_{i=1}^m v_i A_i$, and an associated graph $G(N, E)$ where $E_{mn} = 1$ only if there is some i such that $A_{imn} \neq 0$. Then if G is chordal with some set of maximal cliques \mathcal{C} , we have the following:

Theorem (Agler et al. [4]). *Suppose $G(N, E)$ is chordal. Y is positive semidefinite iff there exist $Z(C_r) \succeq 0$ such that $Y = \sum_{C_r \in \mathcal{C}(G)} Z(C_r)$.*

In addition to improving the SDP solution times, sparse decomposition can also help improve cut performance. Consider a rather extreme case, an LP posed using SDP:

$$\begin{aligned} & \min \text{tr}(\text{diag}(c)X) \\ & \text{subject to} \\ & \text{tr}(\text{diag}(a_i)X) = b_i, \quad \forall i \\ & X \succeq 0 \end{aligned}$$

If we drop the positive semidefinite constraint entirely and apply dense linear cuts, then $d_n(\hat{X})^T X d_n(\hat{X})$ could add a large number of (unbounded) dummy variables. If we instead applied, say, symbolic Cholesky decomposition, then the positive semidefinite constraint would be decomposed using cliques of one node: $X_{mm} \succeq 0 \leftrightarrow X_{mm} \geq 0$. Applying one linear (nonzero) cut to each dense clique ($c_m^2 X_{mm} \geq 0$), we would

recover the LP nonnegativity constraint immediately in the first cutting plane iteration.

Completion and Rank-One for CQCQP

In Chapter 2 we presented CSDP in dense form, but implemented the relaxation in sparse form. This leaves open the question of whether enforcing rank-one constraints on each submatrix of the sparse form retains equivalence with CSDP with rank-one constraint on the dense matrix. We will show that equivalence does indeed hold. Consider a Hermitian matrix X with spectral decomposition $X = \sum_{k=1}^N \lambda_k d_k d_k^*$, where the eigenvalues are ordered so that $\lambda_k \geq \lambda_{k+1}$. Unlike in the real symmetric case, if λ_k has multiplicity 1, then the eigenvector d_k is only unique up to rotation by a complex phase $e^{j\theta_k}$ [154, pp. 41]. In polar coordinates we have that the eigenvector is unique up to scaling of all phase angles by the same degree (preserving angle differences). That is, if $d_{ki} = |d_{ki}|(\cos(\theta_{ki}) + j \sin(\theta_{ki}))$, then we can add $\delta \in \mathbb{R}$ to all angles and replace d_k in the eigenbasis. In terms of rectangular coordinates (i.e. real and imaginary components), we can state that d_k can be replaced in the eigenbasis with

$$\operatorname{Re}(d_k) + \delta \vec{1} + j(\delta^I - \operatorname{Im}(d_k)),$$

for any $\delta \in \mathbb{R}$ such that $(\operatorname{Re}(d_{ki}) + \delta)^2 \leq |d_{ki}|^2 \forall i$ and $\delta^I \in \mathbb{R}^N$ with entries $\delta_i^I = \sqrt{|d_{ki}|^2 - (\delta + \operatorname{Re}(d_{ki}))^2}$.

Sparse positive semidefinite decomposition of a Hermitian matrix $X \in \mathbb{H}^{N \times N}$ yields a set of index sets C , where $X_c \succeq 0 \forall c \in C \iff X \succeq 0$. Note that $\bigcap_{c \in C} c = \{1, \dots, N\}$. A property of sparse decomposition is that C can be represented with an acyclic graph where each node is an element c of C and an edge between two nodes indicates that at least one index is shared between the corresponding index sets; this is known as a clique tree [62, 58].

Proposition 4.3.1. $X_c \succeq 0, \operatorname{rank}(X_c) \leq 1 \forall c \in C$ iff X can be completed so that $X = xx^*$ for some $x \in \mathbb{C}^N$.

Proof. If $X = xx^*$, then $\operatorname{rank}(X) \leq 1, X \succeq 0$ and so one direction is obvious. Now consider the other direction: suppose that $X_c \succeq 0, \operatorname{rank}(X_c) \leq 1 \forall c \in C$. We will use a constructive proof, i.e. we shall construct an x so that $(xx^*)_c = X_c \forall c \in C$.

Let us consider the clique tree corresponding to the chordal graph formed by C . Label a terminal node c_1 . Since X_{c_1} has rank one and is positive semidefinite, we have that $X_{c_1} = \lambda_1^{c_1} (d_1^{c_1})^* d_1^{c_1}$, and so we can set $x_{c_1} = \sqrt{\lambda_1^{c_1}} d_1^{c_1}$ for any normalized principal eigenvector $d_1^{c_1}$. Now denote a neighbouring node c_2 and consider its corresponding index set with some normalized principal eigenvector, $d_2^{c_2}$. By clique tree

property, X_{c_1} and X_{c_2} share at least one entry, say X_{mm} , so $|(d_1^{c_1})_m| = |(d_2^{c_2})_m|$. Since eigenvectors of the eigenbasis are only unique up to rotation by complex phase, d_2 can be rotated to form \hat{d}_2 so that $(d_1^{c_1})_m = (\hat{d}_2^{c_2})_m$, i.e. rotating the eigenvector so that one entry attains a specific angle. Then we can set $x_{c_2} = \sqrt{\lambda_1^{c_2}} \hat{d}_2$, where x_{mm} , the shared entry of x_{c_1}, x_{c_2} retains the same value. The remaining elements of x can be found by proceeding through neighbours in the same manner, with the acyclic property ensuring that each element of x is set once. \square

This relies on a generalization of the fact that in ACOPF and in load flow the bus angles of any solution can be scaled up or down by constants. From the proposition it immediately follows that in the alternative rank condition only the 2×2 principal minors related to the submatrices X_c need to be considered, and so valid inequalities (2.6a) and (2.6b) can be applied in a sparse fashion.

4.4 Computational Experiments

All experiments herein were performed with a 2.26 dual-core Intel i3-350M processor and 4 GB main memory. Experiments were coded in MATLAB (r2010a, see [107]) with model processing from YALMIP (R20130213, see [102]). Test instances were taken from MATPOWER (version 4.1, see [172]), which in turn are derived from IEEE power system test cases. In many cases line ratings were not provided, and angle limits were not added. Conic programs were solved with MOSEK (version 6.0, see [9]), and IPOPT (3.11.1, see [161]) was used as a local solver to obtain primal feasible solutions.

Results are divided into two subsections. First, we discuss the effects of sparse decomposition; although this method has been applied to ACOPF already, there are certain insights here that we believe are worth emphasizing as we have used a more intuitive configuration. Second, we present results on solving SDP via a cutting plane algorithm.

Sparsity

We compare time spent in solvers for various relaxations in Table 4.1, with **cases** indicating the number of buses in the problem, and a dash indicates an unsolved problem due to a lack of memory. The relaxations tested are: the SDP relaxation (SDPR), SDP in dual form (DSDPR), SDP in sparse form (SpSDPR), and the 2×2 principal minor SOCP relaxation (SOCP). SDPR, due to its use of $4N^2$ variables, runs into memory limitations past 57 buses. DSDPR solves faster due to the natural

tendency of dual-form SDPs to use fewer variables. The sparse formulation, denoted SpSDPR, solves substantially faster, but minor convergence issues from this method meant that rank-one solutions could not be obtained directly. Overall formulation time was substantially higher (30 minutes for the 300 bus case), but to some degree this was due to our choice of modeling language, so we have included only time spent in the MOSEK solver.

SOCPR solves considerably faster, as it uses fewer variables and sparser conic constraints. Memory usage was significantly lower as well; for instance, after pre-processing and factorization, for case300 SpSDPR has 2×10^5 nonzero entries in the constraint data matrix, while SOCPR uses 6×10^4 and 4×10^5 for the 2383-bus case. Thus, while the sparse method for SDPR is effective on small instances, SOCPR becomes significantly faster and less memory intensive for medium and large-scale systems. Although SOCPR provides a weaker bound, we can see in Table 4.2 that it is still of high quality. The voltage magnitudes and real power generation schedules are close to the feasible optimum, but reactive power and angles are significantly mismatched, a phenomenon that also occurs in cases with duality gap (see [95]). To some extent this strong performance of SOCPR and SDPR can be explained by the relative ease of the problem; even the copper plate model (no transmission network) provides a good lower bound. This is a property of the IEEE cases, which tend to have ample generation and very little congestion.

The differences between SDPR, its sparse form SpSOCPR, and SOCPR depend on the topology of the underlying network. For dense, highly meshed networks we should expect SpSDPR to take longer to solve, and possibly for the optimal value of SOCPR to diverge from that of SDPR. However, the test problems we have available are rather sparse. For instance, in case300, SpSDPR used less than 5% of the variables of SDPR; the minimum necessary amount could be even less, since we have used a heuristic that satisfies a sufficient condition. In Table 4.3 we show the distribution of clique sizes for the larger instances; clique sizes correspond to a PSD constraint on a submatrix of twice that size. For all instances, more than 80% of cliques involved 2 to 4 buses, indicating a consistent sparsity regardless of problem size. Indeed, cliques of size 2 imply constraints that can be represented exactly with SOCP, so the sparse topology helps explain the strong performance of SOCPR in Table 4.2. Larger cliques appeared sporadically in the largest instances, with the Polish 2383-bus system having a 25-bus clique.

Cuts for a Positive Semidefinite Constraint

We implement a simple cutting plane algorithm: the initial relaxation is solved without PSD enforcement, and in subsequent iteration cuts are added and the relaxation

Table 4.1: Solution times (s)

case	SDPR	DSDPR	SpSDPR	SOCPR
case14	1.06	2.5	0.7	0.1
case30	13.16	0.8	0.81	0.1
case39	55.4	1.9	0.71	0.1
case57	657.74	2.5	1.2	0.1
case118	-	18.2	1.61	0.2
case300	-	363.1	6.47	0.6
case2383wp	-	-	-	7.3

Table 4.2: Relaxation Optimum as Percent of Best Primal Optimum

case	Copper Plate	SOCPR
case14	94.6%	99.9%
case30	98.4%	99.8%
case39	98.6%	99.98%
case57	98.2%	99.9%
case118	97.1%	99.8%
case300	98.1%	99.9%
case2383wp	94.6%	99.3%

Table 4.3: Clique Size Distribution

Clique Size	case118	case300	case2383wp
2	8%	32%	28%
3	61%	36%	53%
4	25%	20%	8%
5	6%	8%	4%
6	0%	3%	2%
7	0%	1%	1%
8 to 25	0%	0%	4%

is solved again. At each iteration we added a cut for each clique-induced positive semidefinite constraint. We tested three types of cuts: a set of linear cuts with $c = [d_i(\hat{X})], \forall i : \lambda_i < 0$ (**lincut**), and the second-order cone cuts **soc1** and **soc2**.

In Table 4.4 we see how quickly cuts can capture the 2×2 complex valid inequalities starting from no enforcement of the PSD condition. The rows are labelled as follows. **case** is the IEEE case name, indicating the number of buses. 2×2 **gap** is the optimality gap between the 2×2 relaxation of ACOPF and the best known solution. **ite** is the number of cutting plane iterations, where each iteration involves adding a set of cuts and solving the modified relaxation. **cuts** is the total number of cuts added. **gap** is the optimality gap between the lower bound established by the cutting plane algorithm and the best known solution. An asterisk indicates failure to converge — either tailing off in convergence or numerical instability causing failure in the interior point method solver. **time** is the total time spent in the relaxation solver; overhead is omitted to remove noise in the comparison. The table indicates difficulty in a cold-start (no PSD limit enforced) approach to solving the SDP relaxation. **soc2** has moderately more success in achieving convergence. **soc1** and **soc2** combined achieve the fastest convergence rate where possible.

Since the 2×2 relaxation performs well, we examine in Table 4.5 the effects of adding PSD cuts to this relaxation. 5 iterations of the cutting plane algorithm were applied. **egap** indicates the optimality gap between the lower bound established by cuts and the best known upper bound. **cgap** is the percent of the 2×2 gap closed by the cuts. The second-order cone cuts performed better than the linear cuts, requiring a similar amount of solving time but closing more gap. **soc1** did not add to the performance of **soc2**. With respect to lower bound quality, the experiment indicates that a few rounds of cuts can bring a 2×2 SOCP relaxation quite close to the SDP relaxation, which has zero gap on these instances.

Table 4.4: Capturing the Strength of 2x2 Complex PSD Inequalities

case	2×2 gap	lincut				soc1				soc2				soc1 + soc2			
		ite	cuts	gap	time	ite	cuts	gap	time	ite	cuts	gap	time	ite	cuts	gap	time
case9	0.0%	21	100	0.6%*	3.5	7	30	0.0%	1.2	5	22	0.0%	0.9	4	36	0.0%	0.8
case14	0.1%	21	241	0.1%	2.5	5	56	0.0%	0.4	4	44	0.0%	0.3	3	64	0.0%	0.3
case30	0.6%	23	586	0.5%	16.4	7	128	0.3%	3.9	3	68	0.2%	1.7	2	90	0.6%	1.4
case57	0.1%	27	1420	98.6*	16.4	49	1102	100%*	37.8	12	523	0.0%	4.6	11	1038	20.7%*	6.8
case118	0.3%	11	1326	99.5%*	17.1	30	3093	6.5%*	52.9	13	1342	7.0%*	13.5	15	3102	10.0%*	30.3
case300	0.2%	13	3461	98.4%*	46.8	13	2649	17.7%*	47	9	1911	3.1%*	30.7	11	4198	18.6%*	54

Table 4.5: Adding PSD Cuts to the 2×2 Relaxation

case	2×2 gap	lincut				soc1				soc2				soc1+soc2			
		cuts	egap	cgap	time	cuts	egap	cgap	time	cuts	egap	cgap	time	cuts	egap	cgap	time
case9	0.0%	20	0.00%	0%	1.2	10	0.00%	66%	1.2	12	0.00%	94%	1.2	16	0.00%	92%	1.2
case14	0.1%	52	0.00%	97%	1.2	31	0.00%	100%	1.1	26	0.00%	100%	1.0	36	0.00%	100%	1.1
case30	0.6%	111	0.53%	8%	5.6	81	0.49%	15%	4.1	73	0.00%	100%	4.4	96	0.00%	100%	4.6
case57	0.1%	254	0.06%	7%	5	203	0.06%	14%	5.7	188	0.01%	89%	5.9	336	0.01%	85%	5.7
case118	0.3%	631	0.25%	4%	11.6	408	0.25%	4%	10.4	362	0.05%	79%	10.8	626	0.05%	80%	11.7
case300	0.2%	1056	0.12%	27%	43	933	0.12%	29%	46.3	916	0.04%	78%	46	1784	0.07%	61%	47.7

4.5 Conclusion

We derived two novel separation rules for second-order cone valid inequalities for a positive semidefinite constraint on a Hermitian matrix. These are closed-form solutions that use the eigenvectors of a matrix. The cuts were applied in sparse fashion to obtain a lower bound for ACOPF problem instances. Computational experiments on IEEE demonstrate that these conic inequalities can be more effective than adding linear inequalities. Further experiments are needed to see if there is a useful ordering of adding cuts, e.g. adding one round of `soc2`, then one round of `soc1` and finishing with `lincut`. There may also be merit in exploring cases where the 2×2 relaxation is substantially worse than the SDP relaxation. Computational results from Kocuk, Dey, and Sun [86] indicate that linear cuts can perform substantially better than the SDP formulation on large cases (2000+) buses. Thus experiments on larger cases may also be worthwhile.

Chapter 5

Unit Commitment with Quadratic Costs

5.1 Introduction

Unit commitment is an electric generation scheduling paradigm where one must determine which generators should be committed to turn on or off at future time periods in order to reliably satisfy consumer demand at minimum cost. The complexities stem from engineering considerations of certain generators such as large thermal units, which require minimum amounts of time to economically and safely start or stop generation. Unit commitment involves further complexities such as forecasting consumer demand, unpredictable generation from renewable sources of energy, and various engineering requirements to reliably operate a power system. Problems are solved using simplifying assumptions so that the unit commitment can be completed within a practical amount of time. Nonlinear constraints are particularly troublesome, and so they are often simplified; for example, modeling alternating current can result in prohibitively difficult problems [57], so linearized power flow is used instead (e.g. [67]). The combinatorial nature of the underlying scheduling problem is such that even a basic linear UC model is NP-Complete [23]. Linearization may be costly as extra generators are turned on to protect against problems that might arise due to the simplifications. Effective methods for nonlinear scheduling problems would allow for more realistic modeling and, in turn, would reduce the costs and possibly the risks of delivering electricity to the consumer.

In this chapter we consider a unit commitment problem with convex quadratic costs. We show that a reformulation can improve solution times of the resulting mixed-integer convex formulation. Although not known at the time this research

was conducted, the idea was already proposed by Günlük and Linderöth [65] under the purview of strengthening formulations with the perspective function. However, no computational experiments were conducted on the problem, so this chapter does present substantial contribution.

5.2 Formulation

A basic formulation of the Unit Commitment for thermal generation follows, where operation costs are minimized subject to meeting demand and generator restrictions:

Data

For each i^{th} generator in the set of generators $\{1, \dots, I\}$ we have:

- MC_i - marginal cost of generating a unit of power
- N_i - fixed per period cost of leaving the generator on
- SU_i - startup cost
- SH_i - shutdown cost
- R_i - ramping limit
- K_i^-, K_i^+ - minimum burn and maximum capacity, respectively
- n_i, f_i - minimum uptime and downtime, respectively

Finally, for each period t in the set of periods $\{1, \dots, T\}$ we have the demand l_t .

Variables

For each generator i and time period t we have:

- $q_{i,t}$ - amount of power generated
- $u_{i,t}$ - indicator that is 1 if generator is turned on in a given period, 0 otherwise
- $s_{i,t}$ - indicator that is 1 if the generator started up in a given period, 0 otherwise
- $h_{i,t}$ - indicator that is 1 if the generator shut down in a given period, 0 otherwise

Model

$$\min_{q,u,s,h} \sum_{i,t} (MC_i q_{i,t} + N_i u_{i,t} + SU_i s_{i,t} + SH_i h_{i,t})$$

Subject to

$$\sum_{i \in I} q_{i,t} = l_t \quad \forall t \in T \quad (5.1a)$$

$$K_i^- u_{i,t} \leq q_{i,t} \leq K_i^+ u_{i,t} \quad \forall i \in I, t \in T \quad (5.1b)$$

$$-R_i \leq q_{i,t} - q_{i,t-1} \leq R_i \quad \forall i \in I, t \in T \quad (5.1c)$$

$$\sum_{\tau=t-n_i+1}^t s_{i,\tau} \leq u_{i,t} \quad \forall i \in I, t \in T \quad (5.1d)$$

$$\sum_{\tau=t-f_i+1}^t h_{i,\tau} \leq 1 - u_{i,t} \quad \forall i \in I, t \in T \quad (5.1e)$$

$$s_{i,t} \geq u_{i,t} - u_{i,t-1} \quad \forall i \in I, t \in T \quad (5.1f)$$

$$h_{i,t} \geq u_{i,t-1} - u_{i,t} \quad \forall i \in I, t \in T \quad (5.1g)$$

$$u_{i,t}, s_{i,t}, h_{i,t} \in \{0, 1\} \quad \forall i \in I, t \in T \quad (5.1h)$$

$$q_{i,t} \geq 0 \quad \forall i \in I, t \in T \quad (5.1i)$$

Constraint (5.1a) ensures that demand is met in all periods. Constraint (5.1b) ensures that production falls between the minimum burn and maximum capacity of each generator. Constraint (5.1c) enforces the ramping limits for each generator, which are technical limitations on the rate of change in generation over time. Constraints (5.1d)-(5.1e) are minimum up-time and minimum down-time constraints, respectively. That is, for maintenance and cost reasons, once turned on a generator must stay on for a minimum amount of time; once turned off, a generator has a minimum cool-down time. The structure from constraints (5.1d)-(5.1e) couple together multiple time periods, making the unit commitment a nontrivial scheduling problem. Constraints (5.1f)-(5.1g) define the startup and shutdown states, respectively, and constraints (5.1h)-(5.1i) enforce decision variable restrictions.

Many variants and extensions of this basic model have been proposed. A significant concern is uncertainty in demand and renewable generation, so stochastic [36, 138] and robust optimization models [171] have been proposed. There are also models incorporating AC transmission systems (e.g. [103, 57]), though these are not yet computationally tractable at industrial scale. Incorporating linearized DC

approximations of the transmission network can approximate network effects; for example, the idea of dispatching transmission – so-called transmission switching – could greatly reduce losses [68, 121]. In electricity markets, issues of fairness and transparency arise, so the choice of solution methods have wider-reaching consequences than strictly minimizing system costs (see [148]). For markets where generators commit their own units, the generation company faces financial risk. Thus several models have been developed to account for this price uncertainty [164, 167, 98, 119, 74].

Numerous methodologies have been proposed for solving the unit commitment, many of them heuristic in nature. In early times, generators were solved using a greedy method called Priority Listing, which takes the lowest average cost units in order until demand can be met. Since this does not guarantee feasibility, numerous ad hoc corrections were to be made by the operators. Lagrangian relaxations became a dominant methodology, where by relaxing the demand constraint, units are decoupled, allowing each generator decision to be easily solved by dynamic programming. Other methods include Benders’ Decomposition [36], MIP (e.g. [66]), and meta-heuristics (e.g. [77, 129]). More details on these methods can be found in surveys (e.g. [143, 126]). There has been a recent trend in favour of MIP, which can provide exact solutions in theory, and in practice offers good lower bounds. For example, the world’s largest wholesale electricity market, PJM, reported savings of \$90M one year after switching from Lagrangian Relaxation methods to Mixed-Integer Programming [152].

5.3 Quadratic Costs and Conic Reformulation

Marginal costs for thermal generators can be represented by quadratic or cubic approximations that capture the nonlinear relationship between fuel and heating [165]. We consider a convex quadratic objective, replacing the linear objective in the previous basic unit commitment model:

$$\min_{q,u,s,h} \sum_{i,t} (MC2_i q_{i,t}^2 + MC_i q_{i,t} + N_i u_{i,t} + SU_i s_{i,t} + SH_i h_{i,t})$$

Subject to (5.1a)-(5.1i).

MC2 denotes the quadratic cost coefficient. The above problem is a Mixed-Integer Quadratic Programming (MIQP) problem, and can be readily solved by a commercial MIP solver such as CPLEX [43]. However, it is possible to build a stronger formulation, using the following result from Aktürk, Atamtürk, and Gürel [6]:

Let $C = \{(x, y, z) \in \mathbb{R}^2 \times \{0, 1\} : y \geq x^{\frac{a}{b}}, uz \geq x \geq lz, x \geq 0\}$, with $a \geq b > 0$ and $u \geq l \geq 0$. The convex hull of C is given by

$$\{(x, y, z) \in \mathbb{R}^3 \times \{0, 1\} : y^b z^{a-b} \geq x^a, uz \geq x \geq lz, 1 \geq z \geq 0, x \geq 0, y \geq 0\}.$$

We can reformulate the quadratic Unit Commitment problem and observe that C is contained in the structure. First we linearize the objective with a substitute variable v :

$$\min_{q,u,s,h} \sum_{i,t} (MC2_i v_{i,t} + MC_i q_{i,t} + N_i u_{i,t} + SU_i s_{i,t})$$

Subject to

$$q_{i,t}^2 \leq v_{i,t} \quad \forall i \in I, t \in T \quad (5.2a)$$

and (5.1a) – (5.1i).

Constraint 5.2a is a rotated second-order cone constraint, which can be represented with second-order cone programming, so the problem is now a mixed-integer conic program. In general, such an approach will not improve solution times, but we observe that constraints (5.1b), (5.1h)-(5.1i), and (5.2a) give the form of C . We can therefore develop the strengthened conic reformulation using the aforementioned description of the convex hull of C :

$$\min_{q,u,s,h} \sum_{i,t} (MC2_i v_{i,t} + MC_i q_{i,t} + N_i u_{i,t} + SU_i s_{i,t})$$

Subject to

$$q_{i,t}^2 \leq u_{i,t} v_{i,t} \quad \forall i \in I, t \in T$$

and (5.1a)-(5.1i), and (5.2a).

As mentioned in the introduction, this has been proposed by Günlük and Linderoth [65] in the context of perspective reformulation, though no computational experiments were conducted in the paper. Other approaches to this problem have involved linear approximations of the objective [55, 53] or solving the MIQP directly [134].

5.4 Computational Experiments

Experiments are performed using CPLEX 12.2 on a 2.3 GHz i3-350M processor with 4GB memory. Generator data and demand are based on the commonly used

Table 5.1: Unit Commitment with Varying Quadratic Cost Coefficients

coef	90% Quad				50% Quad			
	100	100 Conic	50	50 Conic	100	100 Conic	50	50 Conic
rgap	2.61%	-0.01%	2.62%	-0.02%	5.37%	0.2%	5.29%	0.18%
egap	2.21%	-0.01%	1.99%	-0.02%	4.58%	0.2%	4.37%	0.18%
iub	120	40	37	17	713	49	46	19
nodes	19510	0	64658	0	12064	0	33576	0
coef	10% Quad				1% Quad			
	100	100 Conic	50	50 Conic	100	100 Conic	50	50 Conic
rgap	2.29%	1.54%	2.32%	1.52%	1.99%	1.92%	1.84%	1.76%
egap	1.26%	0.65%	1.21%	0.53%	0.67%	0.61%	0.52%	0.52%
iub	1250	56	76	28	1115	45	102	22
nodes	9647	15842	25669	15482	10486	18566	23274	30749

benchmark of Kazarlis, Bakirtzis, and Petridis [77]. The quadratic costs are varied in order to test the effects of the severity of nonlinearity in the objective function. For each quadratic cost setting, the standard formulation and conic reformulation are compared using 50 and 100 generators. Each instance is terminated after 1 hour or if the optimality gap was less than 0.5% before 1 hour. This is a standard optimality gap for the problem (e.g. [54]).

All results are contained in Table 5.1. The first row **coef** denotes the average percent of costs at maximum generation that can be attributed to the quadratic term. Thus 90% Quad has much higher quadratic cost coefficients than 1% Quad. In practice, marginal costs are dependent on fuel costs and generator types, and quadratic curves can vary significantly. The second row **gens** denotes the generator and formulation type. For example, 100 denotes a 100-generator instance using the MIQP formulation, and 100 Conic denotes the conic reformulation of the same instance. Initial root gaps, **rgap** are calculated using the objective of the best known feasible solution, Z^* , and the initial relaxation objective, R : $\text{initial gap} = (Z^* - R) / Z^*$. The end gap **egap** is calculated in the same manner using the best known upper bound at termination. Negative values for gaps are due to numerical issues, but represent a near-optimal solution. **iub** denotes the time to find the initial upper bound via a feasible integer solution. **nodes** are the number of branch-and-bound nodes explored before termination.

As expected, the strengthened conic reformulation results in smaller root gaps. The difference in relaxations is especially pronounced for the highly nonlinear cases. At 90% and 50% quadratic costs, the initial solution was proved to be optimal for the conic reformulation, but gaps in the MIQP formulation were poor. Even for 10%

and 1% quadratic costs, we see that the conic reformulation resulted in significantly faster times to the first feasible solution.

5.5 Conclusion

We developed a strong conic formulation of the unit commitment problem with quadratic costs, which involves more variables than the natural mixed-integer quadratic formulation, but is a tighter relaxation. The experiments showed that conic reformulation resulted in greatly improved relaxations compared to the linear relaxation of the MIQP, particularly when the quadratic coefficient was higher, increasing the curvature of the objective function with respect to generation. In these highly non-linear instances, the initial feasible integer solutions were found to be optimal. In all cases, the first feasible solution was found much faster with the conic reformulation.

Chapter 6

On Mixed-Integer Geometric Programming

6.1 Introduction

We consider the Mixed-Integer Geometric Programming (MIGP) problem:

$$\begin{aligned} & \text{minimize } cx + qy \\ & \text{subject to } e^{a_i x + g_i y - b_i} \leq d_i x + k_i y - h_i, \quad i = 1, 2, \dots, k \\ & \quad x \in \mathbb{Z}^n, y \in \mathbb{R}^p \end{aligned}$$

MIGP is a generalization of geometric programming (GP). GP was developed by Zener [168] in the context of engineering design problems, and its name comes from the use of the arithmetic geometric-mean inequality. Applications include structural design, statistical physics, growth modeling, circuit design, and problems in communication systems (see [41, 33]). Many such applications are naturally formulated in the nonconvex standard form:

$$\begin{aligned} & \text{minimize } f_0(x) \\ & \text{subject to } f_i(x) \leq 1, i = 1, \dots, m \\ & \quad h_j(x) = 1, j = 1, \dots, q \end{aligned}$$

h_i is termed a monomial function, which is of the form $cx_1^{a_1}x_2^{a_2}\dots x_n^{a_n}$ with $c > 0$, and f_i is a posynomial, the sum of such monomials. By change of variables $y_i := \log(x_i)$ such a problem can be put in convex form, as each monomial can be replaced

with $e^{a^T y + \log(c)} = 1 \iff a^T y + \log(c) \geq 0$ and the posynomial constraints may be rewritten as convex sum of exponential constraints.

In this chapter we adopt the exponential inequality of Chandrasekaran and Shah [37], which generalizes geometric programming as the right hand side is not required to be constant. These constraints are convex, and the continuous or natural relaxation where $x \in \mathbb{R}^n$ can be solved in polynomial time with interior point methods provided constraint qualification, e.g. the existence of a strict interior point. Note that a polynomial time interior point algorithm can be constructed even with the addition of second-order cone, and symmetric positive semidefinite conic constraints [37].

Unlike its continuous counterpart, MIGP has received relatively little attention. A generic approach is to treat MIGP as a convex mixed-integer problem and outer-approximate the nonlinear constraints with gradient cuts in order to leverage mixed-integer linear programming algorithms (e.g. [49]). In this chapter we compare this linearization approach with methods that account for the specific structure of MIGP, namely the following mixed-integer set:

$$G := \{x \in \mathbb{Z}_+^n, y \in \mathbb{R}_+^p : e^{ax+gy-b} \leq dx + ky - h\}$$

We explore some possibilities of solving MIGP using Mixed-Integer Second-Order Cone Programming (MISOCP), a class of problems for which there are special-purpose commercial branch-and-cut solvers. To our knowledge, there are no branch-and-bound solvers for MIGP that solve the continuous geometric programming relaxation for lower bounds. We adapt a method to approximate the exponential function in order to outer-approximate MIGP with MISOCP. We also consider submodular cuts, which we will show are applicable to G when x is a binary vector.

6.2 Outer-Approximation of the Exponential Constraint

Glineur [59] proposes using the following limit:

$$e^{-x} = \lim_{\alpha \rightarrow \infty} \left| 1 - \frac{x}{\alpha} \right|^\alpha$$

Each exponential constraint can be approximated with the following convex constraint:

$$\left| 1 + \frac{a_i x + g_i y - b_i}{\alpha} \right|^\alpha \leq d_i x + e_i y - h_i$$

Moreover, Glineur provides the following bounds for $0 \leq x \leq \alpha$:

$$\left|1 - \frac{x}{\alpha}\right|^\alpha \leq e^{-x} < \left|1 - \frac{x}{\alpha}\right|^\alpha + \frac{1}{\alpha}$$

In this case the approximation is an underestimator and hence we have a valid inequality for MIGP that may be used for convex outer-approximation.

Ben-Tal and Nemirovski [20, Ch. 3, pp. 134] combine the aforementioned limit with the Maclaurin series characterization $e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!}$. Define

$$\phi_{p,r}(x) := \left(1 + \frac{x}{2^r} + \frac{1}{2} \left(\frac{x}{2^r}\right)^2 + \dots + \frac{1}{p!} \left(\frac{x}{2^r}\right)^p\right)^{(2^r)}$$

For every $p \geq 1$ we have:

$$\begin{aligned} \lim_{r \rightarrow \infty} \phi_{p,r}(x) &= \lim_{q \rightarrow \infty} \left(1 + \sum_{n=1}^p \frac{x^n/n!}{q^n}\right)^q \\ &= \lim_{q \rightarrow \infty} (\phi_{p,0}(x/q))^q \\ &= \exp\left(\lim_{q \rightarrow \infty} q \ln(\phi_{p,0}(x/q))\right) \\ &= \exp\left(\lim_{q \rightarrow \infty} -q^2 \frac{\frac{d}{dq} \phi_{p,0}(x/q)}{\phi_{p,0}(x/q)}\right) \text{ (L'Hôpital's rule)} \\ &= \exp\left(\frac{x + \lim_{q \rightarrow \infty} \sum_{n=2}^p q^{1-n} x^n / (n-1)!}{1 + \lim_{q \rightarrow \infty} \sum_{n=1}^p \frac{x^n/n!}{q^n}}\right) \\ &= e^x. \end{aligned}$$

For fixed r , the limit on p also converges:

$$\begin{aligned} \lim_{p \rightarrow \infty} \phi_{p,r}(x) &= \left(\sum_{n=0}^{\infty} \frac{\left(\frac{x}{2^r}\right)^n}{n!}\right)^{(2^r)} \\ &= (e^{(x/(2^r))})^{(2^r)} \\ &= e^x \end{aligned}$$

By design, for any positive integers p, r , $\phi_{p,r}$ provides an approximation that is representable with second-order cone constraints [20]. We will give an explicit representation in the next subsection. Ben-Tal and Nemirovski show that for fixed p and $|x| \leq T$ the approximation error $|e^x - \phi_{p,r}(x)| \leq \epsilon$ reduces exponentially:

$$\begin{aligned}
r &\geq O(\ln(\frac{1}{\epsilon}) + T) \\
&\implies (1 - \epsilon)e^x \leq \phi_{p,r}(x) \leq (1 + \epsilon)e^x, |x| \leq T
\end{aligned}$$

Note that $\phi_{p,r}(x)$ can be used to construct a second-order cone representable minorant or underestimator of e^x over a domain $L \leq x \leq U$. Observe that for $x \geq 0$:

$$\begin{aligned}
1 + \frac{x}{2^r} + \frac{1}{2}\left(\frac{x}{2^r}\right)^2 + \dots + \frac{1}{p!}\left(\frac{x}{2^r}\right)^p &\leq \exp\left(\frac{x}{2^r}\right) \\
\iff \phi_{p,r}(x) &\leq e^x
\end{aligned}$$

Hence for a bounded problem $x \geq L$, the constraint $e^x \leq t$ implies the valid inequality $\phi_{p,r}(x - L) \leq e^{-L}t$. This may be viewed as an outer-approximation that is tangent to the true function only when $x = L$.

We note that any MISOCP can outer-approximated with a polyhedral relaxation [47]. However, a more sophisticated approach possible involving extended reformulation (see [21, 160]) may allow for superior performance compared to generic application convex gradient-based cuts. Hence one advantage of relaxing the exponential constraint using SOCP is the possibility of generating better polyhedral relaxations.

Selecting p and r for ϕ

The greater the domain of $x - L$, the more curvature there is to be approximated, and so a higher degree approximation is needed at the same level of approximation error. Let us define the approximation error at a given point as $1 - \phi(x - L)/\exp(x - L)$. Over the range $(x - L, x + U - L)$, ϕ is tangent to \exp at the origin and increases in approximation error as x . We illustrate in Table 6.1 the worst-case approximation error for a given range, specified by the upper bound of $x - L$. `deg` indicates the degree of the polynomial, i.e. $p2^r$. Low-degree approximations are effective for $x - L < 1$. For a given degree polynomial, the series expansion $\phi_{p,0}$ is necessarily the most accurate configuration, i.e. $\phi_{p,0} \geq \phi_{k,r}$ pointwise over the domain of $x - L$ for $2kr \leq p$. However, this configuration of ϕ may require more variables and constraints to formulate using SOCP. Without loss of generality let us consider the constraint $e^x \leq t_0$, where $x \in \mathbb{R}_+$. Then we can represent the outer-approximation with $\phi_{p,r}$ as follows:

Table 6.1: Exponential Approximation Error (percent)

	deg	Upper bound of x-L			
		0.01	0.1	1	10
$\phi_{1,0}$	1	0.00	0.47	26.42	99.95
$\phi_{1,1}$	2	0.00	0.24	17.23	99.84
$\phi_{1,2}$	4	0.00	0.12	10.19	99.32
$\phi_{2,0}$	2	0.00	0.02	8.03	99.72
$\phi_{2,1}$	4	0.00	0.00	2.86	98.45
$\phi_{3,0}$	3	0.00	0.00	1.90	98.97
$\phi_{2,2}$	8	0.00	0.00	0.86	91.25
$\phi_{4,0}$	4	0.00	0.00	0.37	97.07
$\phi_{5,0}$	5	0.00	0.00	0.06	93.29
$\phi_{4,1}$	8	0.00	0.00	0.03	80.60
$\phi_{6,0}$	6	0.00	0.00	0.01	86.99
$\phi_{6,4}$	96	0.00	0.00	0.00	0.01

$$\begin{aligned}
t_i &\geq t_{i+1}^2, & i = 0, \dots, r-1, \\
t_r &= 1 + x/2^r + \sum_{k=2}^p q_{k-1}/k!, \\
q_1 &\geq (x/2^r)^2, \\
q_k(x/2^r) &\geq q_{k/2}^2, & 1 < k \leq p, k \text{ is even}, \\
q_k &\geq q_{(k-1)/2}^2, & 1 < k \leq p, k \text{ is odd}.
\end{aligned}$$

Here $t \in \mathbb{R}_+^r, q \in \mathbb{R}_+^p$ are auxiliary variables for the limit form and Maclaurin series expansion, respectively. Each variable q_k is associated with an additional rotated second-order cone constraint or a convex quadratic constraint, and each variable t_i is associated with an additional convex quadratic constraint. Thus a $\phi_{1,r}$ -approximation uses r variables and constraints, but the equivalent 2^r -degree Maclaurin series expansion uses exponentially more constraints and variables.

6.3 Submodular Inequalities

Definition 6.3.1. A set function $h : 2^N \rightarrow \mathbb{R}$ is *submodular* on N if

$$h(S) + h(T) \geq h(S \cup T) + h(S \cap T) \quad \forall S, T \subseteq N$$

Ahmed and Atamtürk [5] consider set utility functions of the form

$$h(S) = f \left(\sum_{i \in S} a_i \right), S \subseteq N,$$

where $f : \mathbb{R} \rightarrow \mathbb{R}$ is strictly concave, increasing, and differentiable. They identify a condition in which the function is submodular:

Proposition 6.3.1. *The set function h is submodular if $a \geq 0$ or $a \leq 0$.*

Thus when $a \geq 0$ or $a \leq 0$ and $x \in [0, 1]^n$, then $-e^{ax}$ is submodular.

Nemhauser and Wolsey [115] study the problem of maximizing an arbitrary submodular function, which is NP-Hard. They show that the constraint $t \leq h(S)$, $t \in \mathbb{R}$, where h is submodular, can be reformulated using t , $|N|$ binary variables and an exponential number of mixed-integer linear constraints. The polyhedral constraints can be added via cutting planes, for which they provide a greedy separation heuristic.

When the function $h(S) = f \left(\sum_{i \in S} a_i \right)$, $S \subseteq N$ is submodular, Ahmed and Atamtürk exploit the concavity of f and develop stronger valid inequalities than those of Nemhauser and Wolsey. This structure can be exploited in any exponential constraint where x is a binary vector, a has nonzero entries and g is zeros, i.e. $e^{ax-b} \leq dx + ky - h$. We can rewrite such a constraint as follows:

$$\begin{aligned} t &\leq e^b(dx + ky - h) \\ t &\geq e^{ax} \end{aligned}$$

With complementing binary variables q , the second constraint can be transformed as follows:

$$\begin{aligned} t &\leq e^b(dx + ky - h) \\ -e^{-\bar{1}a^+} t &\leq -e^{-a^+q+a^-x^-} \end{aligned}$$

The second constraint will yield a submodular maximization cut due to Proposition 6.3.1, and it will be a valid inequality of the form $t \geq \pi_0 + \pi x$ after substituting out the complementing variables.

Partitioning and Scaling for the 0-1 Mixed-Integer Set

Let us now consider the case where g may have nonzero entries, and let us continue assuming that x is a binary vector. Partition the indices of a into R nonempty subsets $S_i, 1 \leq i \leq R$. Define $a(S_i)$ to be the coefficients of a corresponding to the index set S_i , and likewise let $x(S_i)$ be the corresponding decision variable vector. Moreover, let k be the least integer such that $2^k > R$. Then we may rewrite the mixed 0 – 1 exponential constraint as follows:

$$\exp((gy - b)/2^k) \leq t_0 \quad (6.1a)$$

$$t_0^{2^k} \leq (dx + ky - h) \prod_{i=1}^R t_i \quad (6.1b)$$

$$-\exp(-a(S_i)x(S_i)) \leq -t_i, \quad 1 \leq i \leq R \quad (6.1c)$$

$$t_i \geq 0, \quad 1 \leq i \leq R \quad (6.1d)$$

Constraint (6.1b) may be reformulated using at most 2^{k-1} rotated second-order cone inequalities [7]. The advantage of this approach is that the integer components may be represented with second-order conic constraints, and eq. 6.1a is scaled so that (in the bounded case) the domain may be small enough to use conic outer-approximation. In the special case that $R = n$, each nonconvex constraint in eq. 6.1c may be linearized with a single constraint; otherwise, linearization can be obtained via cuts for submodular minimization (see [13]).

Unfortunately linearization results in a weaker formulation compared to using the mixed-integer exponential constraint. That is, since $-\exp$ is strictly concave, the facets of the convex hull of integer points of $-\exp(-a(S_i)x(S_i))$ are strictly interior to the curve of $-\exp$ at fractional values and therefore underestimates it. We can improve the conic relaxation by scaling:

$$\begin{aligned} \exp[(gy - b + ax)/2^v] &\leq q \\ q^{2^v} &\leq dx + ky - h \end{aligned}$$

As $v \in \mathbb{Z}_{++}$ increases, then $a/2^v$ approaches zero, so the linear underestimation becomes more accurate.

6.4 Application: Capital Budgeting

We consider an application used by Ahmed and Atamtürk [5] as an example of submodular utility function maximization. This is a capital budgeting problem, where the objective is to maximize exponential utility:

$$\max \left\{ \sum_{i=1}^m \pi_i (1 - \exp(-\frac{v_i x}{\lambda})) : ax \leq 1, x \in \{0, 1\}^N \right\}$$

We have N investment options and m scenarios, each with a probability of occurrence π_i and payout vector $v_i \in \mathbb{R}^N$. λ is the parameter for risk aversion, and a is the vector of scaled capital requirements for each investment.

This problem can be reformulated as a pure 0-1 instance of MIGP. We can calculate lower bounds for each scenario's utility with the simple greedy value $L_i = \max_j \frac{v_{ij}}{\lambda a_j}$.

Computational Experiments

Experiments were conducted with CPLEX 12.6 on a single thread of an 8-core Intel i7 2.93 GhZ processor and 8 GB of RAM. Branch-and-bound termination conditions settings were: 0.01% optimality gap, 1 hour time limit, and 1 GB memory limit for the branch-and-bound tree.

First we replicate the data generation procedure of Ahmed and Atamtürk [5]. We compare three methods: standard outer-approximation gradient cuts for each exponential term, submodular valid inequalities for each exponential term, and solving the linear outer-approximation from $\phi_{1,0}$. Table 6.2 shows that the gradient method converged much faster than the submodular cuts, and that the $\phi_{1,0}$ -approximation problem were similarly easy to solve. Matching the results of Ahmed and Atamtürk, all instances with submodular inequalities terminated early due to the memory condition. Table 6.3 compare upper and lower bounds produced by each method. Root gaps were calculated as $\text{rgap} = (z^b - z^r)/z^b$, where z^r is the root relaxation optimum, and z^b the best known upper bound. End gaps were calculated as $\text{egap} = (z^b - z^e)/z^b$, where z^e is the best upper bound at termination. Upper bound gaps were calculated as $\text{ugap} = (z^b - z^u)/z^b$, where z^u is the exponential utility of the incumbent solution found by each method. Note that $\phi_{1,0}$ is a relaxation of the capital budgeting problem, so its lower bounds also bound the original problem's optimal objective value. Results from Table 6.3 indicate that these instances can be solved solely by linear approximation – higher-order convex approximations are not needed. Furthermore, the $\phi_{1,0}$ relaxation only requires a single linear inequality for each exponential function,

Table 6.2: Replicated Capital Budgeting Instances: B&B data

Problem			Gradient Cuts			$\phi_{1,0}$	
n	m	λ	time	nodes	cuts	time	nodes
25	1	1	0.01	64	6	0	45
		2	0.01	62	6	0	45
		4	0	57	6	0	41
	25	1	0.02	112	150	0.01	90
		2	0.02	95	200	0.01	102
		4	0.03	85	200	0	53
	100	1	0.05	85	500	0.01	95
		2	0.04	69	500	0.01	80
		4	0.02	57	500	0.01	68
50	1	1	0.01	65	6	0.01	40
		2	0.02	61	6	0	40
		4	0.01	54	6	0	34
	25	1	0.04	124	250	0.02	42
		2	0.06	189	225	0.02	117
		4	0.05	140	275	0.02	80
	100	1	0.07	131	500	0.02	77
		2	0.06	86	500	0.01	66
		4	0.06	102	500	0.02	41
100	1	1	0.01	21	5	0	31
		2	0.01	19	5	0.01	22
		4	0.01	27	5	0	0
	25	1	0.01	0	125	0	0
		2	0.02	0	125	0	0
		4	0.04	10	125	0	0
	100	1	0.09	104	700	0.01	8
		2	0.08	84	500	0.02	0
		4	0.07	43	400	0.01	0

whereas the gradient approach uses considerably more outer-approximation cuts. The small root gaps demonstrate that some instances of MIGP can be easily solved with a linear method.

Table 6.3: Replicated Capital Budgeting Instances: Gaps

Problem			Gradient Cuts			$\phi_{1,0}$		
n	m	λ	rgap	egap	ugap	rgap	egap	ugap
25	1	1	0.14	0.01	0	0.21	0.06	0
		2	0.06	0.01	0	0.1	0.02	0
		4	0.04	0.01	0	0.05	0.01	0
	25	1	0.11	0.01	0	0.57	0.46	0.01
		2	0.04	0.01	0	0.17	0.12	0
		4	0.02	0.01	0	0.06	0.04	0
	100	1	0.19	0	0	0.72	0.63	0
		2	0.06	0.01	0	0.2	0.17	0
		4	0.03	0.01	0	0.06	0.05	0
50	1	1	0.12	0	0	0.32	0.17	0
		2	0.07	0.01	0	0.12	0.04	0
		4	0.04	0.01	0	0.04	0.02	0
	25	1	0.17	0.01	0	0.61	0.55	0
		2	1.42	0.01	0	0.19	0.15	0
		4	0.03	0.01	0	0.04	0.04	0
	100	1	0.09	0.01	0	0.66	0.59	0
		2	0.05	0.01	0	0.2	0.16	0
		4	0.03	0.01	0	0.07	0.05	0
100	1	1	0.03	0	0	0.13	0.07	0
		2	0.03	0.01	0	0.05	0.02	0
		4	0.02	0.01	0	0.01	0.01	0
	25	1	0.01	0.01	0	0.72	0.71	0
		2	0	0	0	0.19	0.19	0
		4	0.01	0	0	0.05	0.05	0
	100	1	0.03	0	0	0.73	0.72	0
		2	0.02	0.01	0	0.19	0.19	0
		4	0.02	0.01	0	0.06	0.06	0

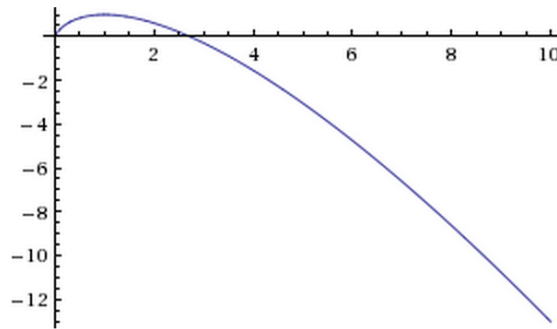
Increasing Root Gap

The capital budgeting problem is solved easily, so we seek to increase the root gap. Consider a simple one-dimensional problem:

$$\min_x e^{bx} - ax, x \in \{0, 1\}$$

The optimal objective is $z^* = \min\{1, e^b - a\}$. If $x^R := \frac{\log(a/b)}{b} \in (0, 1)$, then the optimal relaxation objective is $\frac{a}{b}(1 - \log(\frac{a}{b}))$, otherwise it is z^* . The function $y * (1 - \log(y))$ is shown in Figure 1. It has a global maximum at $y = 1$, and has no other critical points.

Figure 6.1: $y * (1 - \log(y))$



Case 1: $a/b < 1$. This possible iff $a, b < 0$ (so that $x^R > 0$). Now $x^R = \frac{\log(a/b)}{b}$, and so for a given a , b can take an arbitrarily low value and still have $x^R \in (0, 1)$. Thus if $b < 0$, as is the case for capital budgeting, it is desirable to decrease b relative to a as much as possible (optimal integer objective will remain at 1). In the generated instances, the bang-per-buck ratio is close to unity, explaining low root gaps. Root gaps increase as lambda (risk aversion) decreases, which is intuitive as this is equivalent to decreasing b .

Case 2: $a/b > 1$. From the iff observation of Case 1, this means that $a, b > 0$ (a set cover-type problem). Consider two subcases.

Subcase A: $z^* = 1 \iff 1 + a < e^b$. From the graph, root gap will increase as b decreases. Therefore, b should approach $\log(1 + a)$. The calculation is cumbersome, but root gap is monotonically increasing in a when $b = \log(1 + a)$. Thus, arbitrary root gap can be achieved by increasing a .

Subcase B: $1 + a > e^b$. The relative root gap is $1 + \frac{a/b(\log(a/b)-1)}{e^b - a}$. If $e^b - a > 0$, then we are again minimizing on the graph, and this is the same as Subcase A (denominator remains constant when $b = \log(1 + a)$). If $e^b < a$, then for more than 100% root gap we need $\log(a/b) - 1 < 0 \iff a < eb$. But this is impossible as $eb > e^b$ has no solution, so the root gap is bounded by 100%.

High Root Gap Computations

We intend to construct more challenging computational instances with higher root gap and requiring more branching so we can get a better understanding of when (or if) ϕ -approximation should be used instead of gradient cuts.

Consider the following problem, motivated by the previous subsection:

$$\min_{x \in \{0,1\}} \sum_j e^{b_j x} / m - ax : cx \geq 1$$

It is a convex knapsack problem. c is drawn from uniform $[0.9 * 3/n, 1.1 * 3/n]$, so that roughly a third of the variables will be chosen. The objective data is parameterized by α, β . For $j < n$, the j th scenario will draw b_{jj} from Uniform $(0.9\alpha, 1.1\alpha) \times \beta$; all other entries from $v_j(1 - \beta)$, which is a weighted factor of the original Ahmed and Atamtürk data. For positive values of α (and hence b) we will set $a_i = \beta(\exp(\alpha) - 1)/m$ to attain high root gap; otherwise we will set $a_i = -0.1\beta/m$ and reverse the sign of the constraint to $cx \leq 1$. If $\beta = 1$, then maximum root gap is obtained; indeed the relaxation will be so bad that virtually all 2^n possibilities must be explored. The simple variable bound of $L_j := \min b_j/c_j$ is used for each exponential ϕ -approximation valid inequality; this bound is worse as β increases. Some small test cases in Table 6.4 demonstrate that the root gap can be easily made overwhelming. The severe case requires branching over virtually all possibilities (15 choose 5 \approx 3000), as the natural relaxation is rendered ineffective.

Table 6.5 contains results comparing various ϕ configurations with the gradient cuts, **grad**. **nodes** is the number of search tree nodes explored, and **cuts** is the number of gradient cuts included or outer-approximation cuts in the case of ϕ . **time** the total time spent by the solver, and **mem** the maximum memory used during search in MB. **rgap** is calculated as $\text{rgap} = (\text{gub} - \text{rlb}) / |\text{gub}|$, where **gub** is the best known upper bound with respect to the true objective function and **rlb** is the root node lower bound with respect to the given outer-approximation strategy. **egap** is the end gap, calculated in the same manner as **rgap** but using the global lower bound established at termination instead of the root lower bound. We note that the same upper bound solution was found by all methods — hence the difference lies in lower bound quality, time and memory.

More cuts are used as the degree of ϕ increases. $\phi_{2,2}$ achieves the same end gap as the gradient cuts, so an approximation of degree 8 is sufficient for practical purposes here. The substantially large root gap of the gradient cuts is caused by tailing off behaviour in the cutting plane algorithm, resulting in early termination without completely ensuring root node convergence. Furthermore, CPLEX includes some but not all user cuts; for instance in our example we have 600/718 cuts applied. The

Table 6.4: Gradient cuts on small instances

m	n	α	β	nodes	gradcuts	rgap
15	15	1	0.05	160	85	29%
15	15	1	0.5	1374	225	36%
15	15	10	0.05	1705	285	36%
15	15	10	0.5	4137	255	174%

Table 6.5: $\alpha = 1, \beta = 0.5, m = 25, n = 25$

alg	nodes	cuts	time	mem	rgap	egap
grad	63407	600	47	127	18.11%	0.01%
$\phi_{1,0}$	296	-	0.0	0	5.36%	5.25%
$\phi_{1,1}$	4345	276	2	0	3.56%	3.08%
$\phi_{1,2}$	16459	554	13	2	2.66%	1.69%
$\phi_{2,0}$	103790	341	25	6	2.06%	0.91%
$\phi_{2,1}$	209163	655	121	11	1.82%	0.28%
$\phi_{3,0}$	164876	537	75	8	1.75%	0.13%
$\phi_{2,2}$	367497	1098	323	27	1.73%	0.01%
$\phi_{4,0}$	185702	695	98	13	1.75%	0.10%
$\phi_{5,0}$	245724	852	167	19	1.75%	0.09%

substantially higher memory requirements of the gradient cuts indicates relatively higher density. As explained in Subsection 6.2, our implementation of $\phi_{p,r}$ uses $p + r - 1$ additional variables, $r + p/2$ convex quadratic constraints, and $p/2$ rotated conic quadratic constraints. For the budgeting application we have m applications of ϕ .

6.5 Application: System Reliability Redundancy

The series system reliability redundancy problem has been studied since the 1960's (see [89, 131]) and continuous geometric programming relaxations were proposed soon after, starting with Federowicz and Mazumdar [51]. Chern [40] proved that the problem is NP-hard. To our knowledge the problem has not been solved as a mixed-integer geometric program.

Consider a problem with S stages in series. The system fails if there is failure at any stage, and we define the *reliability level* of the system as its probability of success. The probability of failure at a stage s is the probability that all components

in the stage fail. There are J_s components that can be purchased for each stage. Let q_{sj} be the probability of failure of the j th component failing at stage s and let x_{sj} be a binary decision variable used to select the component j . Supposing that failures occur independently, the probability of stage failure is $\prod_{\forall j} q_{sj}^{x_{sj}}$.

Given some cost c_{sj} and minimum reliability level $1 - \epsilon$, we can formulate the problem of minimizing system procurement cost as follows:

$$\begin{aligned} & \min \sum_s \sum_j c_{sj} x_{sj} \\ & \text{subject to } \prod_{\forall s} (1 - \prod_{\forall j} q_{sj}^{x_{sj}}) \geq 1 - \epsilon \\ & \quad x_{sj} \in \{0, 1\} \quad \forall s, j \end{aligned} \tag{6.2a}$$

Introducing auxiliary variables w , we can rewrite the problem as a MIGNP:

$$\begin{aligned} & \min \sum_s \sum_j c_{sj} x_{sj} \\ & \text{subject to } \sum_s w_s \geq \log(1 - \epsilon) \end{aligned} \tag{6.3a}$$

$$\exp(w_s) \leq 1 - t_s \quad \forall s \tag{6.3b}$$

$$\exp\left(\sum_j \log(q_{sj}) x_{sj}\right) \leq t_s \quad \forall s \tag{6.3c}$$

$$x_{sj} \in \{0, 1\} \quad \forall s, j$$

Constraints (6.3b)-(6.3c) enforce $\exp(w_s) \leq 1 - \exp(\sum_j \log(q_{sj}) x_{sj}) = 1 - \prod_{\forall j} q_{sj}^{x_{sj}}$. Therefore, we have that constraint (6.3a) is the log transform of constraint (6.2a), and so there is equivalence between the two problems.

Computational Experiments

For the computational experiments we use the MIP solver of CPLEX Version 12.6 that solves linear outer-approximations of conic quadratic relaxations at the nodes of the branch-and-bound tree. The solver time limit has been set to 3600 seconds, the memory limit to 1 GB, and the search tree nodes limit to 10^6 . All experiments are performed on a 2.93GHz Pentium Linux workstation with 8GB main memory on single-thread processing.

Random instances were generated with varying number of stages S , and components per stage J_s . Each stage was given the same number of possible components J .

Each failure probability q_{sj} was drawn independently and uniformly from $[0.01, 0.2]$. Stage costs c_s were drawn uniformly from $[1, 4]$, and components costs were calculated as $c_{sj} = c_s(-\log(2q_{sj}) + \epsilon_{sj})$, where ϵ is a random variable drawn from $\text{Normal}(0, 0.04)$. The reliability level is controlled by parameter R , with $1 - \epsilon = R^m$.

We compare conic outer-approximation with the gradient method. For constraint (6.3b), we applied either conic outer-approximation (denoted $\phi_{p,r}$) or a standard gradient linearization (**gr**). A practical lower bound could not be found for constraint (6.3c), so we only apply gradient linearization, or linearization together with submodularity cuts. For instance, using the naive solution of x equal to all ones, we have a lower bound of $\sum_j \log(q_{sj})$. This is an extremely pessimistic lower bound; for instance, in the smallest case we will use this is roughly a domain of $-20, 0$ for the exponential function. Moreover, the minimum cost solution tends to be towards the upper bound, so a good approximation is needed. In practical tests, a sufficiently high degree resulted in too much numerical stability to solve the conic approximation.

Results are shown in Table 6.6. Column headings are as follows. **R,S,J** are the problem parameters. **sub** indicates whether submodular cuts were applied. (6.3b) indicates the type of cut applied to constraint (6.3b), and likewise for (6.3c). **nodes** is the number of search tree nodes. **time** is the total time spent in the solver. If convergence to near-optimality did not occur, then either (M) for memory limit or (N) for nodes limit is indicated. The root gap (**rgap**) is calculated as $\text{rgap} = (\text{gub} - \text{rlb}) / |\text{gub}|$, where **gub** is the best known upper bound with respect to the true objective function and **rlb** is the root node lower bound with respect to the given cut strategy. The end gap (**egap**), is calculated as $\text{egap} = (\text{gub} - \text{glb}) / |\text{gub}|$, where **glb** is the global lower bound established at termination. **ub** is the best known upper bound with respect to the true objective function; an asterisk denotes an infeasible solution, which may be possible when the conic outer-approximation is insufficiently tight. **cone** is the number of outer-approximation cuts CPLEX applies to the conic formulations. **cplex** is the number of mixed-integer cuts applied by CPLEX. **user** is the total number of gradient and submodular cuts applied by CPLEX.

As a general trend, the problem became more difficult as reliability, number of stages, or the number of components increased. The pure linear cut strategies performed poorly, although submodular cuts resulted in substantial improvements in convergence irrespective of configuration. Memory limit problems were observed in large problems. $\phi_{4,0}$ had superior performance and was tight enough to produce feasible and near-optimal solutions to the problem with the addition of submodularity cuts.

Table 6.6: Reliability Instances

R	S	J	sub	(6.3b)	(6.3c)	nodes	time	rgap	egap	cone	cplex	user
0.98	10	10	N	gr	gr	14138	2.7	76.34%	0.01%	-	0	316
				$\phi_{4,0}$	gr	1199	0.3	3.92%	0.01%	45	17	40
			Y	gr	gr	10678	2.6	47.67%	0.01%	0	0	384
				$\phi_{4,0}$	gr	652	0.2	1.15%	0.01%	98	10	99
	20		N	gr	gr	26568	4.1	76.19%	0.01%	-	0	247
				$\phi_{4,0}$	gr	4158	1.0	6.36%	0.01%	45	19	45
			Y	gr	gr	26847	5.8	50.48%	0.01%	0	0	439
				$\phi_{4,0}$	gr	15948	3.6	18.84%	0.01%	101	28	110
15	10		N	gr	gr	268984	59.7	76.62%	0.01%	-	0	483
				$\phi_{4,0}$	gr	12551	3.2	4.18%	0.01%	67	26	59
			Y	gr	gr	635714	138.8 (M)	47.85%	1.95%	0	0	529
				$\phi_{4,0}$	gr	11781	3.3	3.02%	0.01%	152	16	141
	20		N	gr	gr	707645	137.4 (M)	76.82%	0.57%	-	0	436
				$\phi_{4,0}$	gr	346722	72.4	5.17%	0.03%	69	23	66
			Y	gr	gr	682210	196.4	50.84%	0.01%	0	0	652
				$\phi_{4,0}$	gr	226565	63.1	9%	0.01%	151	10	177
0.99	10	10	N	gr	gr	407528	79.8	80.02%	0.01%	-	0	340
				$\phi_{4,0}$	gr	178893	36.9	8.67%	0.1%	60	19	56
			Y	gr	gr	162972	36.3	55.74%	0.01%	0	0	429
				$\phi_{4,0}$	gr	25079	6.5	3.5%	0.01%	83	24	137
	20		N	gr	gr	583442	120.9	80.46%	0.01%	-	0	382
				$\phi_{4,0}$	gr	196759	33.5	6.46%	0.03%	63	14	62
			Y	gr	gr	121636	24.9	59.26%	0.01%	0	0	338
				$\phi_{4,0}$	gr	27519	6.2	3.66%	0.01%	83	23	114
15	10		N	gr	gr	488451	89.8 (M)	80.22%	2.01%	-	0	486
				$\phi_{4,0}$	gr	2000000	393.9 (N)	9.79%	0.73%	82	23	87
			Y	gr	gr	416038	99.7 (M)	55.81%	1.48%	0	0	609
				$\phi_{4,0}$	gr	130871	35.3	3.23%	0.96%	137	33	158
	20		N	gr	gr	417958	89.8 (M)	81.03%	4.33%	-	0	681
				$\phi_{4,0}$	gr	2000000	363.9 (N)	6.65%	2.61%	84	14	95
			Y	gr	gr	2000000	393.9 (N)	13.77%	3.67%	28	37	230
				$\phi_{4,0}$	gr	2000000	543.3 (N)	12.82%	2.34%	133	40	243

6.6 Application: Feature Subset Selection for Logistic Regression

Let $a \in \mathbb{R}^n$ be a vector of features or explanatory variables, and let $b \in \{-1, +1\}$ be the associated binary dependent variable. The logistic regression model is given by:

$$\mathbb{P}(b|a) = \frac{1}{1 + \exp(-b(w^T a + c))},$$

where $\mathbb{P}(b|a)$ is the conditional probability of outcome b given $x \in \mathbb{R}^n$. The parameters for this model are the weight vector $w \in \mathbb{R}^n$, and the intercept value $c \in \mathbb{R}$. Given some i.i.d samples $\{a_i, b_i\}_{i=1}^m$, the corresponding likelihood function is $\prod_{i=1}^m \mathbb{P}(b_i|a_i)$. The average logistic loss function l is defined as:

$$\begin{aligned} l(w, c) &:= -1/m \log \prod_{i=1}^m \mathbb{P}(b_i|a_i) \\ &= (1/m) \sum_{i=1}^m \log(1 + \exp(-b_i(w^T a_i + c))) \end{aligned}$$

Thus w, c can be obtained by maximum likelihood estimation, or equivalently by the convex optimization problem of minimizing average logistic loss:

$$\min_{w, c} l(w, c)$$

The solution to this problem can be used to form the logistic classifier $\phi(x) = \text{sgn}(w^T a + c)$, where ϕ identifies the more likely outcome of b given x .

When there are too few observations relative to features, then the standard regression may lead to over-fitting, resulting in a vector w with many nonzero entries (high cardinality) and entries with large magnitudes. A standard approach is to apply regularization, wherein convex penalty functions are added to the objective problem (see [100, 87]). This addresses the problem of large magnitudes directly, and the cardinality of w indirectly.

Sparsity can be induced in w directly via the cardinality constraint $\text{card}(w) \leq k$, which sets an upper bound k on the number of nonzero entries. Selecting the appropriate entries of w to use is the feature subset selection problem (e.g. [114]). The cardinality-constrained logistic regression may be formulated as a MIP problem. Consider the following intermediate formulation of the regression problem:

$$\begin{aligned} & \min(1/m) \sum_{i=1}^m t_i \\ & \text{subject to } \log(1 + \exp(-b_i(w^T a_i + c))) \leq t_i, \quad 1 \leq i \leq m \end{aligned} \quad (6.4a)$$

$$w_j^2 \leq u_j v_j, \quad 1 \leq j \leq n \quad (6.4b)$$

$$\sum_{j=1}^n v_j \leq k \quad (6.4c)$$

$$v \in \{0, 1\}^n$$

v is a vector of binary variables that constrains the support of w , and u, t are auxiliary decision vectors. Introducing auxiliary variables y , we can rewrite this problem as a conic MIGP:

$$\begin{aligned} & \min(1/m) \sum_{i=1}^m t_i \\ & \text{subject to } \exp(-b_i(w^T a_i + c) - t_i) \leq 1 - y_i, \quad 1 \leq i \leq m \end{aligned} \quad (6.5a)$$

$$\exp(-t_i) \leq y_i, \quad 1 \leq i \leq m \quad (6.5b)$$

$$w_j^2 \leq u_j v_j, \quad 1 \leq j \leq n \quad (6.5c)$$

$$\sum_{j=1}^n v_j \leq k \quad (6.5d)$$

$$t \geq 0 \quad (6.5e)$$

$$v \in \{0, 1\}^n$$

Constraints (6.5a)-(6.5b) are equivalent to $\exp(-b_i(w^T a_i + c) - t_i) + \exp(-t_i) \leq 1 \iff 1 + \exp(-b_i(w^T a_i + c)) \leq \exp(t_i)$. Thus constraint (6.4c) can be interpreted as a log transformation, and equivalence is ensured with nonnegativity of t from constraint (6.5e). Constraint (6.5c) is convex and uses the rotated second-order cone, but can be replaced by $L_j v_j \leq w_j \leq U_j v_j$ for sufficiently large bounds. With linear inequalities the problem becomes an instance of MIGP; however, overly large bounds would result in a weak natural relaxation.

Cardinality can be penalized in the objective function instead:

$$\min 2 \sum_{i=1}^m t_i + \lambda \left(\sum_{j=1}^n v_j + 1 \right)$$

$$\text{subject to } \exp(-b_i(w^T a_i + c) - t_i) \leq 1 - y_i, \quad 1 \leq i \leq m \quad (6.6a)$$

$$\exp(-t_i) \leq y_i, \quad 1 \leq i \leq m \quad (6.6b)$$

$$w_j^2 \leq u_j v_j, \quad 1 \leq j \leq n \quad (6.6c)$$

$$t \geq 0 \quad (6.6d)$$

$$v \in \{0, 1\}^n$$

Setting $\lambda = 2$ and $\lambda = \log(n)$ yields a problem of maximizing Aikake (AIC) and Bayesian (BIC) information criteria, respectively (see [140]).

6.7 Conclusion

We considered the mixed-integer geometric programming problem with bounded variables. Conic outer-approximation and submodular cuts can be used on the problem in addition to the standard linear gradient-based outer-approximation applicable to any differentiable convex constraint. We demonstrated with computational experiments on several applications that mixed-integer conic solvers can be used to solve the mixed-integer geometric programming formulations to practical precision.

Chapter 7

Conclusion

7.1 Summary

We developed branch-and-cut algorithms to accommodate a variety of nonlinear nonconvex mathematical programming problems. Chapters 2-4 formed a trilogy focusing on the Alternating Current Optimal Power Flow (ACOPF) problem. In these chapters we constructed a spatial branch-and-cut algorithm for generic quadratically-constrained quadratic programs with complex bounded variables (CQCQP), we developed bound-tightening strategies for power flow constraints, and we introduced a conic outer-approximation method for semidefinite programming (SDP) problems. Chapter 5 demonstrated that the capability of mixed-integer conic solvers by developing a conic reformulation of a mixed-integer convex quadratic programming model of unit commitment. Chapter 6 introduced techniques to solve mixed-integer geometric programs using mixed-integer conic solvers; we applied this to solve a system reliability problem. All these developments were motivated by theory — typically preserving some notion of exactness or algorithm convergence — and their practicality has been confirmed with computational experiments. Thus these are ideas that hold some promise. Whether they last the test of time is another matter, but the hope has been to demonstrate to the reader that branch-and-cut methods are flexible and that global optimization over nonconvex feasible regions is worthy of pursuit.

The remainder of the conclusion is devoted to possible extensions, with applications and methodology discussed separately.

7.2 Extensions: Applications

The efficient inclusion of power flow in problems such as unit commitment would have major benefits; for instance, several undesirable market loopholes could be avoided by accurately accounting for losses and loop flows during generation commitment. Unfortunately the computational experiments suggest that CQCQP may for now be restricted to medium-sized problems, with say hundreds of variables. This is still useful for distribution-level applications (e.g. [157]). Furthermore, instances at this scale can help indicate what is left on the table with regard to current practices, and therefore what is worth pursuing in future research. Moreover, building up a body of knowledge with regards to what can be solved globally is worthwhile in its own right. For instance, in ACOPF there is the folk knowledge that current solvers are adequate, but in the absence of rigorous testing (e.g. high duality gap cases), this verdict remains unscientific. Convincing testing could pave the way to nonlinear nonconvex AC power pricing in real-time markets. This could have numerous benefits such as improved economic valuation of reactive power support.

We have focused on applications in the energy industry, although the problem structures we have studied provide tremendous flexibility to a clever modeler. For instance, nonconvex quadratic constraints can be used to model polynomial constraints, which can model a very wide range of problems. Another example is in mixed-integer geometric programming. The convex exponential constraint has potential applications in machine learning. The field of machine learning has begun to expand its focus from local convexity analysis to structure integer sets, namely submodularity (e.g. [14]). Bertsimas and Shioda [24] demonstrated some of the potential benefits of embracing integer nonconvexity; for instance, they show that endogenous outlier removal can be performed in linear regression. We can, of course, come full circle and apply new machine learning techniques to the energy industry. For instance, the electricity markets have an abundance of high-dimensional data and with constant regulatory updates there remain many questions that may be addressed with statistics (e.g. [97]).

7.3 Extensions: Methodology

CQCQP

There are many avenues to further improve spatial branch-and-cut for CQCQP problems such as ACOPF. For instance, our branching is effectively local, concentrating on 2×2 principal minors, so cuts involving more variables (e.g. [141, 142]) may

be complementary to this approach. This could be seen as analogous to the typical MIP branch-and-cut approach, branching on fractional variables, but employing cuts involving more structural information. Different variable selection strategies more specifically geared to ACOPF can also be explored. Our tightening techniques, in fact, enable a fair amount of flexibility in this regard. For instance, branching on real nodal power would result in the appropriate bound contraction w.r.t. voltage magnitudes, and hence make use of our valid inequalities.

The valid inequalities developed for branch-and-cut involved a somewhat simple set \mathcal{J} , so a natural question is whether it is possible to generalize further. For instance, it would be quite valuable to find an efficient procedure to find pertinent facets of the convex hull of a 2×2 complex PSD rank-1 decision matrix together with an arbitrary number of linear constraints. This would allow us to describe the convex hull of a bus pair while accounting for real and reactive power flows. We demonstrated in our special case that the convex envelope of an eigenvalue constraint (equivalent to the rank constraint) over a certain convex projection coincided with the convex hull of \mathcal{J} . Again, it may be possible to generalize the result, or else variants of the approach may be useful in analyzing other rank-constrained sets. The lifted relaxation of CSDP also seems worthy of further consideration, for unlike in the original space of CQCQP, which seemingly can produce any number of intricate feasible regions, the rank-one constrained CSDP is a set of points along the positive semidefinite cone. As the feasible region of CSDP can be interpreted as the intersection of a polyhedron with the positive semidefinite cone, there may be opportunities to extend geometrically motivated cuts for MIP to CSDP with rank-one constraint.

One could also form various hybrid algorithms using the developments in Chapters 2-4. For instance, bound tightening using full relaxations could be judiciously applied to problematic buses, and our closed-form methods could then be used to propagate bound changes on neighbouring buses. Moreover, further investigation is needed to see if propagating angle bound changes on larger cycles may be useful. It may also be worthwhile to consider bound tightening on other problems involving AC power flow equations, such as the mixed-integer CQCQP optimal capacitor location problem (e.g. [73]). Mixed-integer CQCQP would naturally lend itself to the SDP outer-approximation method, as this would allow the use of mixed-integer linear or second-order cone solvers to be used.

Mixed-Integer Geometric Programming

We studied the integer structure of mixed-integer geometric programming via submodularity. However, another point of view is that of disjunctive programming on convex problems (e.g. [63, 30]). This route may allow the development of convex

valid inequalities and perhaps a better understanding of the exponential inequality with unbounded integer variables. It may also be possible to leverage Mixed-Integer Rounding [115]. The main obstacle is finding a useful aggregation of polyhedral constraints that could make such an approach viable; if we draw on experience with conic integer programming (e.g. [12]), extended formulations are likely in store.

Appendix A

Appendix

A.1 Creating Bounds for CSDP from CQCQP

The diagonal bounds follow directly from the bounds on x , i.e., $L_{kk} = |\ell_k|^2, U_{kk} = |u_k|^2$. The off-diagonal bound is given in ACOPF and is a phase condition that implies that x_1 cannot have a purely real nonzero solution if x_2 has a purely imaginary nonzero solution, and vice versa. In the real case we can set $L_{12} = U_{12} = 0$. However, such a bound can always be derived from CQCQP via transformation of variables. Let $x = w + it$. Since $X = xx^*$ in the lifted formulation of CQCQP we have $W_{12} = w_1w_2 + t_1t_2$ and $T_{12} = t_1w_2 - w_1t_2$. A sufficient condition to derive the off-diagonal bound is that either w or t (or both) have strictly positive entries. We can apply an affine transformation on CQCQP, so this is not a restrictive requirement. Observe that for any ℓ we have:

$$x^*Qx + c^*x + b = (x - \ell)^*Q(x - \ell) + (c^* + 2\ell^*Q)(x - \ell) + b + \ell^*Q\ell + c^*\ell.$$

Therefore with substitution of variables $y := x - \ell + e + ie$, where e is the ones vector, any bounded complex QCQP may be rewritten with a decision vector with only positive components.

Then we can assume $0 < W^- \leq W_{12}$, where $W^- := w_1^L w_2^L + t_1^L t_2^L$. From $\text{rank}(X) = 1$ we have that $W_{12}^2 + T_{12}^2 = W_{11}W_{22}$, and so:

$$\sqrt{\frac{U_{11}U_{22}}{(W^-)^2} - 1} \geq \sqrt{\frac{W_{11}W_{22}}{W_{12}^2} - 1} = \frac{|T_{12}|}{W_{12}}.$$

Hence we have valid inequalities with $-L_{12} = U_{12} = \sqrt{U_{11}U_{22}/(W^-)^2 - 1}$. Note that if the affine transformation was used and the valid inequalities are translated

back to the original space, then the inequalities will also include the original variables of CQCQP. For instance, suppose we set $y = x - l + e + \iota e$. Moreover, define the components $y := w^y + \iota t^y$. Then we have

$$\begin{aligned}
W_{ij}^y &= (w_i - w_{ii}^L + 1)(w_j - w_j^L + 1) + (t_i - t_{ii}^L + 1)(t_j - t_j^L + 1) \\
&= w_i w_j + t_i t_j + (1 - w_j^L)w_i + (1 - w_i^L)w_j + (1 - w_i^L)(1 - w_j^L) \\
&\quad + (1 - t_j^L)t_i + (1 - t_i^L)t_j + (1 - t_i^L)(1 - t_j^L) \\
&= W_{ij} + (1 - w_j^L)w_i + (1 - w_i^L)w_j + (1 - t_j^L)t_i + (1 - t_i^L)t_j \\
&\quad + (1 - w_i^L)(1 - w_j^L) + (1 - t_i^L)(1 - t_j^L).
\end{aligned}$$

Thus a valid inequality for W_{ij}^y gives a valid inequality for W_{ij}, x_i, x_j .

Bibliography

- [1] T Achterberg, T Berthold, and G Hendel. “Rounding and propagation heuristics for mixed integer programming”. In: *Operations Research Proceedings 2011*. Springer, 2012, pp. 71–76.
- [2] T Achterberg, T Koch, and A Martin. “Branching rules revisited”. In: *Operations Research Letters* 33 (2005), pp. 42–54.
- [3] T Achterberg et al. “Constraint integer programming: A new approach to integrate CP and MIP”. In: *Integration of AI and OR techniques in constraint programming for combinatorial optimization problems*. Springer, 2008, pp. 6–20.
- [4] J Agler et al. “Positive semidefinite matrices with a given sparsity pattern”. In: *Linear algebra and its applications* 107 (1988), pp. 101–149.
- [5] S Ahmed and A Atamtürk. “Maximizing a class of submodular utility functions”. In: *Mathematical programming* 128 (2011), pp. 149–169.
- [6] M S Aktürk, A Atamtürk, and S Gürel. “A strong conic quadratic reformulation for machine-job assignment with controllable processing times”. In: *Operations Research Letters* 37 (2009), pp. 187–191.
- [7] F Alizadeh and D Goldfarb. “Second-order cone programming”. In: *Mathematical programming* 95 (2003), pp. 3–51.
- [8] O Alsac et al. “Further developments in LP-based optimal power flow”. In: *IEEE Transactions on Power Systems* 5 (1990), pp. 697–711.
- [9] E D Andersen and K D Andersen. “The MOSEK interior point optimizer for linear programming: an implementation of the homogeneous algorithm”. In: *High performance optimization* 33 (2000), pp. 197–232.
- [10] K M Anstreicher. “Semidefinite programming versus the reformulation linearization technique for nonconvex quadratically constrained quadratic programming”. In: *Journal of Global Optimization* 43 (2009), pp. 471–484.

- [11] K M Anstreicher and S Burer. “Computable representations for convex hulls of low-dimensional quadratic forms”. In: *Mathematical programming* 124 (2010), pp. 33–43.
- [12] A Atamtürk and V Narayanan. “Conic mixed-integer rounding cuts”. In: *Mathematical programming* 122 (2010), pp. 1–20.
- [13] A Atamtürk and V Narayanan. “Polymatroids and mean-risk minimization in discrete optimization”. In: *Operations Research Letters* 36 (2008), pp. 618–622.
- [14] F Bach. “Learning with submodular functions: A convex optimization perspective”. In: *arXiv preprint arXiv:1111.6453* (2011).
- [15] X Bai and H Wei. “A semidefinite programming method with graph partitioning technique for optimal power flow problems”. In: *International Journal of Electrical Power & Energy Systems* 33 (2011), pp. 1309–1314.
- [16] X Bai et al. “Semidefinite programming for optimal power flow problems”. In: *International Journal of Electrical Power & Energy Systems* 30 (2008), pp. 383–392.
- [17] X Bao, N V Sahinidis, and M Tawarmalani. “Multiterm polyhedral relaxations for nonconvex, quadratically constrained quadratic programs”. In: *Optimization Methods & Software* 24 (2009), pp. 485–504.
- [18] C Barnhart et al. “Branch-and-price: Column generation for solving huge integer programs”. In: *Operations research* 46 (1998), pp. 316–329.
- [19] P Belotti et al. “Mixed-integer nonlinear optimization”. In: *Acta Numerica* 22 (2013), pp. 1–131.
- [20] A Ben-Tal and A Nemirovski. *Lectures on modern convex optimization: analysis, algorithms, and engineering applications*. Vol. 2. SIAM, 2001.
- [21] A Ben-Tal and A Nemirovski. “On polyhedral approximations of the second-order cone”. In: *Mathematics of Operations Research* (2001), pp. 193–205.
- [22] A Ben-Tal, A Nemirovski, and C Roos. “Extended matrix cube theorems with applications to μ -theory in control”. In: *Mathematics of Operations Research* 28 (2003), pp. 497–523.
- [23] D P Bertsekas et al. “Optimal short-term scheduling of large-scale power systems.” In: *IEEE Transactions on Automatic Control* 28 (1983), pp. 1–11.
- [24] D Bertsimas and R Shioda. “Classification and regression via integer optimization”. In: *Operations Research* 55 (2007), pp. 252–271.

- [25] D Bienstock and A Michalka. “Cutting-planes for optimization of convex functions over nonconvex sets”. In: *SIAM Journal on Optimization* 24 (2014), pp. 643–677.
- [26] D Bienstock and G Munoz. “LP approximations to mixed-integer polynomial optimization problems”. In: *ArXiv e-prints* (2015).
- [27] D Bienstock and G Munoz. “On linear relaxations of OPF problems”. In: *arXiv preprint arXiv:1411.1120* (2014).
- [28] R Bixby and E Rothberg. “Progress in computational mixed integer programming – a look back from the other side of the tipping point”. In: *Annals of Operations Research* 149 (2007), pp. 37–41.
- [29] J R S Blair and B Peyton. “An introduction to chordal graphs and clique trees”. In: *Graph theory and sparse matrix computation*. Springer, 1993, pp. 1–29.
- [30] P Bonami et al. “On mathematical programming with indicator constraints”. In: *Mathematical Programming* 151 (2015), pp. 191–223.
- [31] S Bose et al. “Quadratically constrained quadratic programs on acyclic graphs with application to power flow”. In: *IEEE Transactions on Control of Network Systems* PP (2015).
- [32] S P Boyd and L Vandenberghe. *Convex optimization*. Cambridge Univ Pr, 2004. ISBN: 0521833787.
- [33] S Boyd et al. “A tutorial on geometric programming”. In: *Optimization and engineering* 8 (2007), pp. 67–127.
- [34] S Burer and D Vandembussche. “Globally solving box-constrained nonconvex quadratic programs with semidefinite-based finite branch-and-bound”. In: *Computational Optimization and Applications* 43 (2009), pp. 181–195.
- [35] S Ceria and J Soares. “Convex programming for disjunctive convex optimization”. In: *Mathematical Programming* 86 (1999), pp. 595–614.
- [36] S Cerisola et al. “Stochastic power generation unit commitment in electricity markets: A novel formulation and a comparison of solution methods”. In: *Operations research* 57 (2009), pp. 32–46.
- [37] V Chandrasekaran and P Shah. “Conic geometric programming”. In: *Information Sciences and Systems (CISS), 2014 48th Annual Conference on*. IEEE, 2014, pp. 1–4.

- [38] C Chekuri and S Khanna. “A polynomial time approximation scheme for the multiple knapsack problem”. In: *SIAM Journal on Computing* 35 (2005), pp. 713–728.
- [39] C Chen, A Atamtürk, and S S Oren. “A Spatial Branch-and-Bound Algorithm for Nonconvex QCQP with Bounded Complex Variables”. 2015.
- [40] M-S Chern. “On the computational complexity of reliability redundancy allocation in a series system”. In: *Operations Research Letters* 11 (1992), pp. 309–315.
- [41] M Chiang. *Geometric programming for communication systems*. Now Publishers Hanover, 2005.
- [42] C Coffrin and P Van Hentenryck. “A linear-programming approximation of AC power flows”. In: *INFORMS Journal on Computing* 26 (2014), pp. 718–734.
- [43] IBM ILOG CPLEX. “V12. 1: User’s Manual for CPLEX”. In: *International Business Machines Corporation* 46 (2009), p. 157.
- [44] M Davis. “Hilbert’s tenth problem is unsolvable”. In: *American Mathematical Monthly* (1973), pp. 233–269.
- [45] E De Klerk. “The complexity of optimizing over a simplex, hypercube or sphere: a short survey”. In: *Central European Journal of Operations Research* 16 (2008), pp. 111–125.
- [46] A De Maio et al. “Design of optimized radar codes with a peak to average power ratio constraint”. In: *IEEE Transactions on Signal Processing* 59 (2011), pp. 2683–2697.
- [47] S Drewes and S Ulbrich. “Subgradient Based Outer Approximation for Mixed Integer Second Order Cone Programming”. In: *Mixed Integer Nonlinear Programming* (2012), pp. 41–59.
- [48] M Dür. “Copositive programming—a survey”. In: *Recent advances in optimization and its applications in engineering*. Springer, 2010, pp. 3–20.
- [49] M A Duran and I E Grossmann. “An outer-approximation algorithm for a class of mixed-integer nonlinear programs”. In: *Mathematical programming* 36.3 (1986), pp. 307–339.
- [50] J Edmonds. “Submodular functions, matroids, and certain polyhedra”. In: *Combinatorial structures and their applications* (1970), pp. 69–87.

- [51] A J Federowicz and M Mazumdar. “Use of geometric programming to maximize reliability achieved by redundancy”. In: *Operations Research* 16 (1968), pp. 948–954.
- [52] Samuel Fiorini et al. “Linear vs. semidefinite extended formulations: exponential separation and strong lower bounds”. In: *Proceedings of the forty-fourth annual ACM symposium on Theory of computing*. ACM. 2012, pp. 95–106.
- [53] A Frangioni and C Gentile. “A computational comparison of reformulations of the perspective relaxation: SOCP vs. cutting planes”. In: *Operations Research Letters* 37 (2009), pp. 206–210.
- [54] A Frangioni, C Gentile, and F Lacalandra. “Solving unit commitment problems with general ramp constraints”. In: *International Journal of Electrical Power & Energy Systems* 30.5 (2008), pp. 316–326.
- [55] A Frangioni, C Gentile, and F Lacalandra. “Tighter Approximated MILP Formulations for Unit Commitment Problems”. In: *IEEE Transactions on Power Systems* 24 (2009), pp. 105–113.
- [56] Robert M Freund and Jorge R Vera. “Condition-based complexity of convex optimization in conic linear form via the ellipsoid algorithm”. In: *SIAM Journal on Optimization* 10 (1999), pp. 155–176.
- [57] Y Fu, M Shahidehpour, and Z Li. “Security-constrained unit commitment with AC constraints”. In: *IEEE Transactions on Power Systems* 20 (2005), pp. 1001–1013.
- [58] M Fukuda et al. “Exploiting sparsity in semidefinite programming via matrix completion I: General framework”. In: *SIAM Journal on Optimization* 11 (2001), pp. 647–674.
- [59] F Glineur. “Topics in convex optimization: interior-point methods, conic duality and approximations”. PhD thesis. Polytechnic College of Mons, 2001.
- [60] R E Gomory. “An algorithm for integer solutions to linear programs”. In: *Recent advances in mathematical programming* 64 (1963), pp. 260–302.
- [61] A Gopalakrishnan et al. “Global optimization of optimal power flow using a branch & bound algorithm”. In: *Communication, Control, and Computing (Allerton), 2012 50th Annual Allerton Conference on*. IEEE. 2012, pp. 609–616.
- [62] R Grone et al. “Positive definite completions of partial Hermitian matrices”. In: *Linear algebra and its applications* 58 (1984), pp. 109–124.

- [63] I E Grossmann and S Lee. “Generalized convex disjunctive programming: Nonlinear convex hull relaxation”. In: *Computational Optimization and Applications* 26 (2003), pp. 83–100.
- [64] M Grötschel, L Lovász, and A Schrijver. “The ellipsoid method and its consequences in combinatorial optimization”. In: *Combinatorica* 1 (1981), pp. 169–197.
- [65] O Günlük and J Linderoth. “Perspective reformulations of mixed integer nonlinear programs with indicator variables”. In: *Mathematical programming* 124 (2010), pp. 183–205.
- [66] K W Hedman, R P O’Neill, and S S Oren. “Analyzing valid inequalities of the generation unit commitment problem”. In: *IEEE/PES Power Systems Conference and Exposition, 2009*. IEEE. 2009, pp. 1–6.
- [67] K W Hedman et al. “Co-optimization of generation unit commitment and transmission switching with N-1 reliability”. In: *Power Systems, IEEE Transactions on* 25 (2010), pp. 1052–1063.
- [68] K W Hedman et al. “Optimal transmission switching – sensitivity analysis and extensions”. In: *Power Systems, IEEE Transactions on* 23 (2008), pp. 1469–1479.
- [69] I A Hiskens and R J Davy. “Exploring the power flow solution space boundary”. In: *IEEE Transactions on Power Systems* 16 (2001), pp. 389–395.
- [70] R A Horn and C R Johnson. *Matrix analysis*. Cambridge university press, 1990.
- [71] Y Huang and D P Palomar. “Randomized algorithms for optimal solutions of double-sided QCQP with applications in signal processing”. In: *IEEE Transactions on Signal Processing* 62 (2014), pp. 1093–1108.
- [72] R A Jabr. “Exploiting sparsity in SDP relaxations of the OPF problem”. In: *IEEE Transactions on Power Systems* 27 (2012), pp. 1138–1139.
- [73] R A Jabr. “Optimal power flow using an extended conic quadratic formulation”. In: *IEEE Transactions on Power Systems* 23 (2008), pp. 1000–1008.
- [74] R A Jabr. “Robust Self-Scheduling Under Price Uncertainty Using Conditional Value-at-Risk”. In: *IEEE Transactions on Power Systems* 20 (2005), pp. 1852–1858.
- [75] C Jozs et al. “Application of the moment-SOS approach to global optimization of the OPF problem”. In: *IEEE Transactions on Power Systems* 30 (2013), pp. 463–470.

- [76] N Karmarkar. “A new polynomial-time algorithm for linear programming”. In: *Proceedings of the sixteenth annual ACM symposium on Theory of computing*. ACM. 1984, pp. 302–311.
- [77] S A Kazarlis, A G Bakirtzis, and V Petridis. “A genetic algorithm solution to the unit commitment problem”. In: *Power Systems, IEEE Transactions on* 11 (2002), pp. 83–92.
- [78] J E Kelley Jr. “The cutting-plane method for solving convex programs”. In: *Journal of the Society for Industrial & Applied Mathematics* 8 (1960), pp. 703–712.
- [79] L G Khachiyan. “Polynomial algorithms in linear programming”. In: *USSR Computational Mathematics and Mathematical Physics* 20 (1980), pp. 53–72.
- [80] S Kim, M Kojima, and M Yamashita. “Second order cone programming relaxation of a positive semidefinite constraint”. In: *Optimization Methods and Software* 18 (2003), pp. 535–541.
- [81] S Kim et al. “Exploiting sparsity in linear and nonlinear matrix inequalities via positive semidefinite matrix completion”. In: *Mathematical programming* 129 (2011), pp. 33–68.
- [82] S Kirkpatrick, C D Gelatt, and M P Vecchi. “Optimization by simulated annealing”. In: *science* 220 (1983), pp. 671–680.
- [83] K Kleibohm. “Bemerkungen zum Problem der nichtkonvexen Programmierung”. In: *Mathematical Methods of Operations Research* 11 (1967), pp. 49–60.
- [84] I Klep and M Schweighofer. “An exact duality theory for semidefinite programming based on sums of squares”. In: *Mathematics of Operations Research* 38.3 (2013), pp. 569–590.
- [85] B Kocuk, S S Dey, and X A Sun. “Inexactness of SDP Relaxation for Optimal Power Flow over Radial Networks and Valid Inequalities for Global Optimization”. In: *IEEE Transactions on Power Systems* PP (2014).
- [86] Burak Kocuk, Santanu S Dey, and X Andy Sun. “Strong SOCP Relaxations for Optimal Power Flow”. In: *arXiv preprint arXiv:1504.06770* (2015).
- [87] K Koh, S-J Kim, and S P Boyd. “An interior-point method for large-scale l1-regularized logistic regression.” In: *Journal of Machine learning research* 8 (2007), pp. 1519–1555.
- [88] K Krishnan and J E Mitchell. “A unifying framework for several cutting plane methods for semidefinite programming”. In: *Optimization methods and software* 21 (2006), pp. 57–74.

- [89] W Kuo. *Optimal reliability design: fundamentals and applications*. Cambridge university press, 2001.
- [90] A H Land and A G Doig. “An automatic method of solving discrete programming problems”. In: *Econometrica: Journal of the Econometric Society* (1960), pp. 497–520.
- [91] J B Lasserre. “Global optimization with polynomials and the problem of moments”. In: *SIAM Journal on Optimization* 11 (2001), pp. 796–817.
- [92] J Lavaei and S H Low. “Zero Duality Gap in Optimal Power Flow Problem”. In: *IEEE Transactions on Power Systems* 27 (2012).
- [93] P Le Bodic and G L Nemhauser. “How important are branching decisions: fooling MIP solvers”. In: *Operations Research Letters* 43 (2015), pp. 273–278.
- [94] K Lehmann, A Grastien, and P Van Hentenryck. “AC-Feasibility on Tree Networks is NP-Hard”. In: *IEEE Transactions on Power Systems* PP (2014).
- [95] B C Lesieutre et al. “Examining the limits of the application of semidefinite programming to power flow problems”. In: *Communication, Control, and Computing (Allerton), 2011 49th Annual Allerton Conference on*. IEEE. 2011, pp. 1492–1499.
- [96] F Li and R Bo. “DCOPF-based LMP simulation: Algorithm, comparison with ACOPF, and sensitivity”. In: *IEEE Transactions on Power Systems* 22 (2007), pp. 1475–1485.
- [97] R Li, A J Svoboda, and S S Oren. “Efficiency impact of convergence bidding in the california electricity market”. In: *Journal of Regulatory Economics* (2014), pp. 1–40.
- [98] T Li, M Shahidehpour, and Z Li. “Risk-Constrained Bidding Strategy With Stochastic Unit Commitment”. In: *IEEE Transactions on Power Systems* 22 (2007), pp. 449–458.
- [99] J Linderoth. “A simplicial branch-and-bound algorithm for solving quadratically constrained quadratic programs”. In: *Mathematical programming* 103 (2005), pp. 251–282.
- [100] J Liu, J Chen, and J Ye. “Large-scale sparse logistic regression”. In: *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM. 2009, pp. 547–556.
- [101] M S Lobo et al. “Applications of second-order cone programming* 1”. In: *Linear Algebra and its Applications* 284 (1998), pp. 193–228.

- [102] J Lofberg. “YALMIP: A toolbox for modeling and optimization in MATLAB”. In: *Computer Aided Control Systems Design, 2004 IEEE International Symposium on*. IEEE. 2004, pp. 284–289.
- [103] A Lotfjou et al. “Security-Constrained Unit Commitment With AC/DC Transmission Systems”. In: *IEEE Transactions on Power Systems* 25 (2010), pp. 531–542.
- [104] L Lovász. “Submodular functions and convexity”. In: *Mathematical Programming The State of the Art*. Springer, 1983, pp. 235–257.
- [105] L Lovász and A Schrijver. “Cones of matrices and set-functions and 0-1 optimization”. In: *SIAM Journal on Optimization* 1 (1991), pp. 166–190.
- [106] D G Luenberger and Y Ye. *Linear and nonlinear programming*. Vol. 116. Springer Science & Business Media, 2008.
- [107] The mathworks. “MATLAB User’s Guide”. In: *Inc., Natick, MA* 5 (1998).
- [108] G P McCormick. “Computability of global solutions to factorable nonconvex programs: Part 1 Convex underestimating problems”. In: *Mathematical programming* 10 (1976), pp. 147–175.
- [109] R R Meyer. “On the existence of optimal solutions to integer and mixed-integer programming problems”. In: *Mathematical Programming* 7 (1974), pp. 223–235.
- [110] R Misener and C A Floudas. “GloMIQO: Global mixed-integer quadratic optimizer”. In: *Journal of Global Optimization* 57 (2013), pp. 3–50.
- [111] J E Mitchell. “Polynomial interior point cutting plane methods”. In: *Optimization Methods and Software* 18 (2003), pp. 507–534.
- [112] D K Molzahn and I A Hiskens. “Sparsity-exploiting moment-based relaxations of the optimal power flow problem”. In: *IEEE Transactions on Control of Network Systems* PP (2014).
- [113] D K Molzahn et al. “Implementation of a Large-Scale Optimal Power Flow Solver Based on Semidefinite Programming”. In: *IEEE Transactions on Power Systems* (2013), pp. 1–12.
- [114] P M Narendra and K Fukunaga. “A branch and bound algorithm for feature subset selection”. In: *Computers, IEEE Transactions on* 100 (1977), pp. 917–922.
- [115] G L Nemhauser and L A Wolsey. *Integer and combinatorial optimization*. Vol. 18. Wiley New York, 1988.

- [116] A Nemirovski. “Advances in convex optimization: conic programming”. In: *International Congress of Mathematicians*. Vol. 1. 2006, pp. 413–444.
- [117] Y Nesterov, A Nemirovskii, and Y Ye. *Interior-point polynomial algorithms in convex programming*. Vol. 13. SIAM, 1994.
- [118] Y Nesterov, M J Todd, and Y Ye. “Infeasible-start primal-dual methods and infeasibility detectors for nonlinear programming problems”. In: *Mathematical Programming* 84 (1999), pp. 227–267.
- [119] E Ni, P B Luh, and S Rourke. “Optimal Integrated Generation Bidding and Scheduling With Risk Management Under a Deregulated Power Market”. In: *IEEE Transactions on Power Systems* 19 (2004), pp. 600–609.
- [120] J Nie. “Optimality conditions and finite convergence of Lasserre’s hierarchy”. In: *Mathematical Programming* 146 (2014), pp. 97–121.
- [121] R P O’Neill et al. “Economic analysis of the N-1 reliable unit commitment and transmission switching problem using duality concepts”. In: *Energy Systems* 1 (2010), pp. 165–195.
- [122] M R Oskoorouchi and J E Mitchell. “A second-order cone cutting surface method: complexity and application”. In: *Computational Optimization and Applications* 43 (2009), pp. 379–409.
- [123] A L Ott. “Evolution of computing requirements in the PJM market: Past and future”. In: *2010 IEEE Power and Energy Society General Meeting*. IEEE. 2010, pp. 1–4.
- [124] T J Overbye, X Cheng, and Y Sun. “A comparison of the AC and DC power flow models for LMP calculations”. In: *System Sciences, 2004. Proceedings of the 37th Annual Hawaii International Conference on*. IEEE. 2004, 9–pp.
- [125] M Padberg and G Rinaldi. “A branch-and-cut algorithm for the resolution of large-scale symmetric traveling salesman problems”. In: *SIAM review* 33.1 (1991), pp. 60–100.
- [126] N P Padhy. “Unit commitment-a bibliographical survey”. In: *IEEE Transactions on Power Systems* 19 (2004), pp. 1196–1205.
- [127] C H Papadimitriou. “On the complexity of integer programming”. In: *Journal of the ACM (JACM)* 28 (1981), pp. 765–768.
- [128] G Pataki. “Bad semidefinite programs: they all look the same”. In: *Arxiv preprint arXiv:1112.1436* (2011).

- [129] S Patra, S K Goswami, and B Goswami. “Differential Evolution Algorithm for Solving Unit Commitment with Ramp Constraints”. In: *Electric Power Components and Systems* 36 (2008), pp. 771–787.
- [130] D T Phan. “Lagrangian duality-based branch and bound algorithms for optimal power flow”. In: *Operations Research* 60 (2012), pp. 275–285.
- [131] F Proschan and T A Bray. “Optimum redundancy under multiple constraints”. In: *Operations Research* 13 (1965), pp. 800–814.
- [132] Jakob Puchinger and Günther R Raidl. *Combining metaheuristics and exact algorithms in combinatorial optimization: A survey and classification*. Springer, 2005.
- [133] A Qualizza, P Belotti, and F Margot. “Linear programming relaxations of quadratically constrained quadratic programs”. In: *Mixed Integer Nonlinear Programming* (2012), pp. 407–426.
- [134] R Quan et al. “A two-stage method with mixed integer quadratic programming for unit commitment with ramp constraints”. In: *IEEE International Conference on IEEM, 2008*. IEEE. 2008, pp. 374–378.
- [135] Ulrich Raber. “A simplicial branch-and-bound method for solving nonconvex all-quadratic programs”. In: *Journal of Global Optimization* 13 (1998), pp. 417–432.
- [136] F Rossi, P Van Beek, and T Walsh. *Handbook of constraint programming*. Elsevier, 2006.
- [137] Thomas Rothvoß. “Some 0/1 polytopes need exponential size extended formulations”. In: *Mathematical Programming* 142 (2013), pp. 255–268.
- [138] P A Ruiz, C R Philbrick, and P W Sauer. “Modeling Approaches for Computational Cost Reduction in Stochastic Unit Commitment Formulations”. In: *IEEE Transactions on Power Systems* 25 (2010), pp. 588–589.
- [139] N V Sahinidis. “BARON: A general purpose global optimization software package”. In: *Journal of Global Optimization* 8 (1996), pp. 201–205.
- [140] T Sato et al. “Feature Subset Selection for Logistic Regression via Mixed Integer Optimization”. In: (2015).
- [141] A Saxena, P Bonami, and J Lee. “Convex relaxations of non-convex mixed integer quadratically constrained programs: extended formulations”. In: *Mathematical programming* 124 (2010), pp. 383–411.

- [142] A Saxena, P Bonami, and J Lee. “Convex relaxations of non-convex mixed integer quadratically constrained programs: projected formulations”. In: *Mathematical programming* 130 (2011), pp. 359–413.
- [143] G B Sheble and G N Fahd. “Unit commitment literature synopsis”. In: *Power Systems, IEEE Transactions on* 9 (1994), pp. 128–135.
- [144] H D Sherali and C H Tuncbilek. “A global optimization algorithm for polynomial programming problems using a reformulation-linearization technique”. In: *Journal of Global Optimization* 2 (1992), pp. 101–112.
- [145] H D Sherali and C H Tuncbilek. “A reformulation-convexification approach for solving nonconvex quadratic programming problems”. In: *Journal of Global Optimization* 7 (1995), pp. 1–31.
- [146] N Z Shor. “Quadratic optimization problems”. In: *Soviet Journal of Circuits and Systems Sciences* 25 (1987), p. 6.
- [147] C K Sim and G Zhao. “A note on treating a second order cone program as a special case of a semidefinite program”. In: *Mathematical programming* 102 (2005), pp. 609–613.
- [148] R Sioshansi, R P O’Neill, and S S Oren. “Economic Consequences of Alternative Solution Methods for Centralized Unit Commitment in Day-Ahead Electricity Markets”. In: *IEEE Transactions on Power Systems* 23 (2008), pp. 344–352.
- [149] S Sojoudi and J Lavaei. “Network Topologies Guaranteeing Zero Duality Gap for Optimal Power Flow Problem”. In: *Submitted to IEEE Transactions on Power Systems* (2011).
- [150] S Sojoudi, R Madani, and J Lavaei. “Low-rank solution of convex relaxation for optimal power flow problem”. In: *2013 IEEE International Conference on Smart Grid Communications (SmartGridComm)*. IEEE, 2013, pp. 636–641.
- [151] B Stott, J Jardim, and O Alsac. “DC power flow revisited”. In: *IEEE Transactions on Power Systems* 24 (2009), pp. 1290–1300.
- [152] D Streiffert, R Philbrick, and A Ott. “A mixed integer programming solution for market clearing and reliability analysis”. In: *Power Engineering Society General Meeting, 2005. IEEE*. IEEE, 2005, 2724–2731 Vol. 3.
- [153] D I Sun et al. “Optimal power flow by Newton approach”. In: *IEEE Transactions on Power Apparatus and Systems* (1984), pp. 2864–2880.
- [154] T Tao. *Topics in random matrix theory*. Vol. 132. American Mathematical Soc., 2012.

- [155] F Tardella. “Existence and sum decomposition of vertex polyhedral convex envelopes”. In: *Optimization Letters* 2 (2008), pp. 363–375.
- [156] M Tawarmalani and N V Sahinidis. “Convex extensions and envelopes of lower semi-continuous functions”. In: *Mathematical Programming* 93 (2002), pp. 247–263.
- [157] J A Taylor and F S Hover. “Convex models of distribution system reconfiguration”. In: *IEEE Transactions on Power Systems* 27 (2012), pp. 1407–1413.
- [158] D Vandembussche and G L Nemhauser. “A branch-and-cut algorithm for non-convex quadratic programs with box constraints”. In: *Mathematical Programming* 102 (2005), pp. 559–575.
- [159] A Verma. “Power grid security analysis: An optimization approach”. PhD thesis. Columbia University, 2009.
- [160] J P Vielma, S Ahmed, and G L Nemhauser. “A lifted linear programming branch-and-bound algorithm for mixed-integer conic quadratic programs”. In: *INFORMS Journal on Computing* 20 (2008), pp. 438–450.
- [161] A Wächter and L T Biegler. “On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming”. In: *Mathematical Programming* 106 (2006), pp. 25–57.
- [162] I Waldspurger, A D’Aspremont, and S Mallat. “Phase recovery, maxcut and complex semidefinite programming”. In: *Mathematical Programming* 149 (2015), pp. 47–81.
- [163] H Wolkowicz, R Saigal, and L Vandenberghe. *Handbook of semidefinite programming: theory, algorithms, and applications*. Vol. 27. Springer Science & Business Media, 2012.
- [164] H Y Yamin and S M Shahidehpour. “Risk and profit in self-scheduling for GenCos”. In: *IEEE Transactions on Power Systems* 19 (2004), pp. 2104–2106.
- [165] H Yang, P Yang, and C Huang. “Evolutionary programming based economic dispatch for units with non-smooth fuel cost functions”. In: *Power Systems, IEEE Transactions on* 11 (1996), pp. 112–118.
- [166] M Yannakakis. “Computing the minimum fill-in is NP-complete”. In: *SIAM Journal on Algebraic Discrete Methods* 2 (1981), pp. 77–79.
- [167] Z Yu. “A spatial mean-variance MIP model for energy market risk analysis”. In: *Energy economics* 25 (2003), pp. 255–268.

- [168] C Zener. “A mathematical aid in optimizing engineering designs”. In: *Proceedings of the National Academy of Sciences of the United States of America* 47 (1961), p. 537.
- [169] B Zhang and D Tse. “Geometry of feasible injection region of power networks”. In: *2011 49th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*. IEEE. 2011, pp. 1508–1515.
- [170] W Zhang and R E Korf. “Performance of linear-space search algorithms”. In: *Artificial Intelligence* 79 (1995), pp. 241–292.
- [171] L Zhao and B Zeng. “Robust unit commitment problem with demand response and wind energy”. In: *2012 IEEE Power and Energy Society General Meeting*. 2012, pp. 1–8.
- [172] R D Zimmerman, C E Murillo-Sánchez, and R J Thomas. “MATPOWER: Steady-state operations, planning, and analysis tools for power systems research and education”. In: *IEEE Transactions on Power Systems* 26 (2011), pp. 12–19.