Convex Relaxation Techniques for Nonlinear Optimization

Eltved, Anders

Publication date:
2021

Document Version
Publisher's PDF, also known as Version of record

Convex Relaxation Techniques for Nonlinear Optimization

Anders Eltved

Kongens Lyngby 2020

Technical University of Denmark
Department of Applied Mathematics and Computer Science
Richard Petersens Plads, building 324,
2800 Kongens Lyngby, Denmark
Phone +45 4525 3031
compute@compute.dtu.dk
www.compute.dtu.dk

Summary (in English)

Optimization is everywhere. In science and engineering, optimization is widely used for various applications, and many of the problems that we would like to solve are nonlinear. In general, we do not have efficient methods for solving nonlinear problems, so we have to rely on local optimization methods that provide a candidate solution to our problem with no guarantee of optimality and no bound on the possible suboptimality.

For some hard problems, we can use convex relaxation techniques to approximate the original (hard) problem in a certain way by one that we can solve efficiently. This approximation, which is a convex optimization problem, gives us a bound on the optimal value of the original problem. This bound can be used to gauge the suboptimality of candidate solutions. Bounds on the optimal value also play an important role in branch-and-bound algorithms for hard combinatorial problems. In some cases, the convex relaxation gives us a solution to the original problem.

In this thesis, we are particularly interested in the so-called Shor semidefinite relaxation where the convex optimization problem is a semidefinite program, *i.e.*, a linear program involving a matrix variable that is constrained to be positive semidefinite. This relaxation has proven to be a good approximation for many interesting problems, including the so-called optimal power flow problem, where the goal is to generate and distribute power in a power network at a minimum cost.

The goal of the thesis is to contribute to the understanding of convex relaxation and add to the existing toolbox of techniques. We present a numerical

study that demonstrates that the semidefinite relaxation of the optimal power flow problem can be solved reliably for large power networks in a few minutes. We present a new technique for strengthening the semidefinite relaxation for an extended trust region subproblem which as an extension of the classical trust region subproblem. We present a framework for guaranteeing that the semidefinite relaxation of a specific problem class is exact, which means that the problem can be solved efficiently by solving the convex relaxation.

Summary (in Danish)

Optimering er overalt. Indenfor videnskab og ingeniørkunst er optimering meget udbredt og mange af de problemer vi ønsker at løse er ikkelineære. Generelt har vi ikke effektive metoder til at løse ikkelineære problemer, så vi må nøjes med at bruge lokale optimeringsmetoder, som giver os en mulig løsning uden garanti for at den er optimal og uden en begrænsning på dens potentielle suboptimalitet.

For nogle svære problemer kan vi bruge konvekse relakseringsteknikker til at approksimere det oprindelige problem på en bestemt måde med et problem som vi kan løse effektivt. Denne approksimation, som er et konvekst optimeringsproblem, giver os en begrænsing på den optimale værdi af det oprindelige problem. Denne begrænsning kan bruges til at vurdere hvor suboptimal en mulig løsning er. Begrænsninger af den optimale værdi spiller også en stor rolle i branch-and-bound algoritmer til svære kombinatoriske problemer. I nogle tilfælde giver den konvekse relaksering os en løsning til det oprindelige problem.

I denne afhandling er vi specielt interesserede i den såkaldte Shor semidefinit relaksering, hvor det konvekse optimeringsproblem er et semidefinit programmeringsproblem, hvilket er et lineært optimeringsproblem med en matrix variabel som skal være positiv semidefinit. Denne relaksering har vist sig at være en god approksimation for mange interessante problemer, inklusiv det såkaldte optimal power flow problem, hvor målet er at generere og distribuere strøm i et el-netværk til den laveste pris.

Målet med denne afhandling er at bidrage til forståelsen af konveks relaksering og tilføje teknikker til den eksisterende værktøjskasse. Vi præsenterer et numerisk studie, der demonstrerer at den semidefinitte relaksering af optimal power

flow problemet kan løses pålideligt for større el-netværk på få minutter. Vi præsenterer en ny teknik til at styrke den semidefinitte relaksering af et såkaldt extended trust region subproblem, som er en udvidelse af det klassiske trust region problem. Vi præsenterer en metode til at garantere at den semidefinitte relaksering af en specifik problemklasse er eksakt, hvilket betyder at problemet kan løses effektivt ved hjælp af den konvekse relaksering.

Preface

This thesis was prepared at the Technical University of Denmark (DTU) in partial fulfillment of the requirements for acquiring a PhD degree.

The work has been carried out in the period from 01.01.2017 to 31.12.2020 at DTU Compute in the Section for Scientific Computing. The main supervisor of the PhD project has been Associate Professor Martin S. Andersen and the co-supervisor has been Associate Professor Spyros Chatzivasileiadis.

A part of the work was carried out at the University of Iowa during a research stay from 16.09.2019 to 31.01.2020 hosted by Professor Samuel Burer. The stay was partly funded by Otto Mønsteds Fond.

The project has been funded by a PhD scholarship awarded by DTU Compute.

The aim of the thesis is to summarize and present the work and results detailed in a published paper and two submitted manuscripts. These are referred to as Papers A–C and appear in Appendices A–C. Chapters 1–2 provide an overview of convex relaxation techniques for nonlinear optimization in the perspective of this project. Chapter 3 discusses exactness and summarizes Paper C. Chapter 4 discusses strengthening and summarizes Paper B. Chapter 5 summarizes Paper A and discusses the optimal power flow problem and how the papers relate to this. Chapter 6 concludes the thesis with an outlook on possibilities for future research.

Kongens Lyngby, 31-12-2020

Anders Eltved

Acknowledgments

There are many people that have contributed to this thesis becoming a reality. First and foremost, I am grateful to my advisor Martin for getting me on this path and for all that he has taught me along the way. I am grateful to my co-advisor Spyros for answering all my questions about power systems. I am grateful to my host advisor, Sam, for hosting me with great hospitality and for taking the time—I learned a lot. I am also grateful for all the other teachers and advisors I have had at all institutions since the very beginning.

I am grateful to my family, friends, and colleagues along the way; thank you for making it an enjoyable ride.

I am grateful to Majbritt for being by my side the whole way; thank you for all your support and for always believing in me.

En ph.d. er fordybelsens kunst
Forskningens svar på man kravler
Ét spørgsmål besvaret
Så er den vel klaret
Men, ak; spørgsmål, de avler

Contents

List of Figures

List of Tables

List of Papers

The thesis contains the following papers in Appendices A–C:

A [33] Anders Eltved, Joachim Dahl, and Martin S. Andersen. "On the robustness and scalability of semidefinite relaxation for optimal power flow problems". *Optimization and Engineering* 21.2 (Mar. 2019), pp. 375–392. DOI: 10.1007/s11081-019-09427-4

B [32] Anders Eltved and Samuel Burer. "Strengthened SDP Relaxation for an Extended Trust Region Subproblem with an Application to Optimal Power Flow". *arXiv e-prints*, arXiv:2009.12704 (Sept. 2020), arXiv:2009.12704. arXiv: 2009.12704 [math.OC]

C [31] Anders Eltved and Martin S. Andersen. "Sufficient Conditions for Exact Semidefinite Relaxation of Homogeneous Quadratically Constrained Quadratic Programs with Forest Structure". Submitted. 2020

Chapter 1

Introduction

Optimization is a word that brings many association and is widely used. Broadly speaking, when there is something we want to optimize, we have a goal that we want to achieve and perhaps some things that constrain us. The goal and constraints are usually somewhat fuzzy, but if we can formulate them mathematically, there is a chance that we can get a computer to help us out. Mathematical optimization, where a problem is formulated in a certain mathematical form, is ubiquitous in science and engineering. Applications are numerous and varied; they include, for example, finding an optimal design, optimization of chemical processes, and finding the best placement of wind turbines in a wind farm. Indeed, the demand for optimization is high, and the problems that we wish to solve are increasingly difficult. Our world is complex and, as a consequence, our problems are usually complex too. This means that even though we have a mathematical formulation of our problem, it is not guaranteed that we have an efficient method for solving it.

In this thesis, we explore convex relaxation techniques, where a hard optimization problem is approximated in a certain way by one that is easier to solve. The convex relaxation gives us some information about our original problem and sometimes it even gives us a solution.

In the following we provide some background and describe the challenges that we address in this thesis. At the end of this chapter, we define our notation and

terminology and outline the structure of the thesis.

1.1 Background

A mathematical optimization problem can be formulated as

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & x \in \mathcal{F} \end{array} \tag{P}$$

where the variables are $x \in \mathbb{R}^n$, $f : \mathbb{R}^n \to \mathbb{R}$ is the objective function, and $\mathcal{F} \subseteq \mathbb{R}^m$ is the feasible set. The feasible set is usually described in terms of a number of functions:

$$\mathcal{F} = \{x \in \mathbb{R}^n : g_i(x) \leq 0,\ i = 1, \ldots, m\}, \tag{1.1}$$

where $g_i : \mathbb{R}^n \to \mathbb{R}$ and the inequalities $g_i(x) \leq 0$ are called constraints. In words, the goal is to find an x in $\mathcal{F}$ for which f attains its lowest value. If a point $x^\star \in \mathcal{F}$ satisfies $f(x^\star) \leq f(x)$ for all $x \in \mathcal{F}$, we call $x^\star$ a *minimizer*, or a solution, of problem (P) and we call $f(x^\star)$ the *optimal value*. If we can find such a point, we say that we have solved (P).

The combination of objective function, f, and feasible set, $\mathcal{F}$, determines the complexity of solving (P). We consider a problem efficiently solvable if there exist an algorithm that solves the problem to a predefined numerical accuracy in polynomial time in the size of the problem (number of variables and constraints). For example, when f is linear and $\mathcal{F}$ is polyhedral (all g_i linear), problem (P) is called a *linear program* (LP) and there exist efficient algorithms for solving problems of this form [27, 72]. On the other hand, when f or any g_i is nonlinear, we generally do not have any efficient methods for solving (P) (to global optimality). When f is a nonlinear function or $\mathcal{F}$ is characterized by nonlinear functions, we call (P) a *nonlinear optimization problem*.

For nonlinear problems it is common practice to use methods for computing a *local minimizer*. A local minimizer of (P) is a point x^ℓ which satisfies $f(x^\ell) \leq f(x)$ for all $x \in \{x \in \mathcal{F} : \|x - x^\ell\| \leq \delta\}$ for some $\delta > 0$. In words, a local minimizer is a point that attains the lowest objective value in a neighborhood around itself. Note that a (global) minimizer is also a local minimizer by definition. Nonlinear problems can have many local minimizers and the objective value of these can be different. For a local minimizer, x^ℓ, we refer to the difference in objective value to the optimal value $f(x^\ell) - f(x^\star)$ as a measure of *suboptimality*. We call methods for computing local minimizers *local optimization methods*. In this context, methods for computing a minimizer are often called *global* optimization methods.

In the distinction between local and global optimization, a particularly interesting class of problems is one where the objective, f, is a convex function and the feasible set, $\mathcal{F}$, is convex[1]. We call these problems *convex optimization problems* [18, 11]. The appeal of convex optimization problems comes from the fact that we have efficient (polynomial-time) methods for many classes of convex optimization problems and that any local minimizer is also a global minimizer. Hence, given a local minimizer, one is never left wondering if there exists a better local minimizer. Even when the problem is convex, it is not necessarily efficiently solvable [30]. Problems that can be solved efficiently include conic linear programs (cone LPs) of the form

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & Ax = b \\ & x \in \mathcal{K}, \end{array} \tag{1.2}$$

where the variables are $x \in \mathbb{R}^n$, the data are $A \in \mathbb{R}^{m \times n}$ and $b \in \mathbb{R}^m$, and $\mathcal{K}$ is a direct product of the following convex cones:

- the nonnegative orthant, $R_+^n = \{x \in \mathbb{R}^n : x_i \geq 0,\ i = 1, \ldots, m\}$;
- the second-order cone (SOC), $\text{SOC} = \{(v_0, v) \in \mathbb{R} \times \mathbb{R}^{n-1} : \|v\| \leq v_0\}$;
- the cone of symmetric positive semidefinite matrices, $\mathcal{S}_+^n = \{A \in \mathbb{R}^{n \times n} : A = A^T \ \wedge \ y^T A y \geq 0 \ \forall y \in \mathbb{R}^n\}$.

With convex relaxation techniques we try to use the efficient solvability of convex optimization problems to gain information about a nonlinear problem which cannot be solved efficiently. An optimization problem

$$\begin{array}{ll} \text{minimize} & \tilde{f}(x) \\ \text{subject to} & x \in \mathcal{R} \end{array} \tag{R}$$

is a *relaxation* of problem (P) if $\mathcal{F} \subseteq \mathcal{R}$ and $f(x) \geq \tilde{f}(x)$. In words, the objective function of the relaxation is a global underestimator for the original objective function and the feasible set of the relaxation should contain the original feasible set. An illustration of the relationship between the feasible sets can be seen in Figure 1.1. Note that we can always consider a so-called epigraph formulation of problem (P) to obtain an equivalent problem with a convex objective:

$$\begin{array}{ll} \text{minimize} & t \\ \text{subject to} & (t, x) \in \mathcal{F}_{\text{epi}} \end{array} \tag{1.3}$$

where the variables are $t \in \mathbb{R}$ and $x \in \mathbb{R}^n$ and the feasible set is $\mathcal{F}_{\text{epi}} = \{(t, x) \in \mathbb{R} \times \mathbb{R}^n : x \in \mathcal{F},\ f(x) \leq t\}$. Hence, a relaxation can be seen as a problem

[1] A definition of convexity is given in Section 1.3.

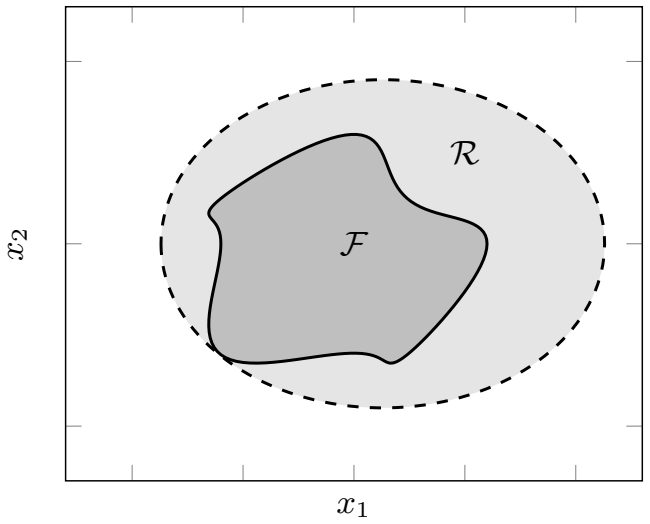


Figure 1.1: Illustration of relationship between the feasible set $\mathcal{F}$ of the original problem and the feasible set $\mathcal{R}$ of the relaxation.

with a larger feasible set in order to avoid the discussion of an underestimator for the objective. However, many of the problems considered in this thesis are reformulated in such a way that we can use the original objective as the objective of the relaxation ($\tilde{f} = f$). Hence, much of the discussion will revolve around the feasible sets $\mathcal{F}$ and $\mathcal{R}$. Note that a relaxation of (1.3) is

$$\begin{array}{ll} \text{minimize} & t \\ \text{subject to} & (t, x) \in \mathcal{R}_{\text{epi}} \end{array} \tag{1.4}$$

where $\mathcal{R}_{\text{epi}} = \{(t, x) \in \mathbb{R} \times \mathbb{R}^n : x \in \mathcal{R},\ \tilde{f}(x) \leq t\}$, which is exactly an epigraph formulation for problem (R).

There are often multiple ways to obtain a relaxation. For example, one could choose $\mathcal{R} = \mathbb{R}^n$ to obtain an unconstrained problem. However, for the relaxation to be useful, it is better to choose an $\mathcal{R}$ that approximates $\mathcal{F}$ well in some sense. This motivates the term *tightness*. The tightness of a relaxation is a measure of how close the relaxation is to the original problem, and it is usually taken to be the difference between their respective optimal values and is referred to as the *relaxation gap*. Denoting the optimal value of (P) by $p^\star$ and the optimal value of (R) by $p_\star$, the relaxation gap can be defined as

$$\text{gap} = p^\star - p_\star. \tag{1.5}$$

This measure requires the optimal value of the original problem, which we generally cannot compute, so in practice a local solution is often used in place of $p^\star$, when we want to evaluate the tightness of a relaxation. When we solve the relaxation (R) to obtain $p_\star$, it may be the case that the minimizer, which we will denote $x_\star$, is in the original feasible set. If this is the case, it is also a minimizer of the original problem (P) and we say that the relaxation is *exact*. The optimal value of the relaxation, $p_\star$, is a lower bound on the optimal value of the original problem, *i.e.*, $f(x) \geq p_\star$ for all $x \in \mathcal{F}$. As a consequence, we can use the relaxation to bound the suboptimality of a local minimizer, x^ℓ, since

$$f(x^\ell) - f(x^\star) \leq f(x^\ell) - p_\star. \tag{1.6}$$

Hence, $f(x^\ell) - p_\star$ is an upper bound on the suboptimality of x^ℓ. A tighter relaxation results in a better bound. A relaxation is most useful, when we have an efficient method for solving (R). Therefore, it is common practice to use relaxations where the problem (R) is a convex optimization problem. When this is the case, we call (R) a *convex relaxation* of (P).

Suppose that the objective function, $\tilde{f}$, for a relaxation is given. If we want a convex relaxation the best choice of feasible set, $\mathcal{R}$, would be to use the convex hull of the original feasible set, which we will denote

$$\mathcal{C} = \overline{\operatorname{conv}}\left\{x : x \in \mathcal{F}\right\}. \tag{1.7}$$

For a set S, $\overline{\operatorname{conv}}\left\{S\right\}$ denotes the closure of the convex hull of S. This is the smallest convex set that contains $\mathcal{F}$. However, we generally do not know a tractable representation of $\mathcal{C}$ although it can be described for some $\mathcal{F}$ [19].

Since a problem can have multiple relaxations, we establish some terminology to compare relaxations. Consider two different relaxations: relaxation 1 with feasible set $\mathcal{R}_1$ and relaxation 2 with feasible set $\mathcal{R}_2$, both with the same objective function. We say that relaxation 1 is stronger than, or dominates, relaxation 2 if $\mathcal{R}_1 \subset \mathcal{R}_2$. This is illustrated in Figure 1.2. For a pair of relaxations with feasible sets $\mathcal{R}_1$ and $\mathcal{R}_3$, it may also be the case that neither of the relaxations is stronger than the other. This is illustrated in Figure 1.3.

When the relaxation (R) of a problem (P) is solved, we will find ourselves in one of three scenarios: (1) the relaxation is exact, *i.e.*, $p^\star = p_\star$ and $x_\star \in \mathcal{F}$[2]; (2) the relaxation is not exact, but provides a lower bound on $p^\star$; (3) the relaxation is infeasible, *i.e.*, $\mathcal{R} = \emptyset$. These scenarios are illustrated in Figure 1.4. In scenarios

[2]We note briefly that it can happen that $p^\star = p_\star$ but $x_\star \notin \mathcal{F}$ if the solution of the relaxation is not unique. In this case, a minimizer, $x^\star$, of (P) is also a minimizer of (R). One might say that the relaxation is exact in this case, but we may not have a way to compute $x^\star$. For the semidefinite programming (SDP) relaxation, described in Section 2.1, this is related to finding low rank solutions of SDPs [57].

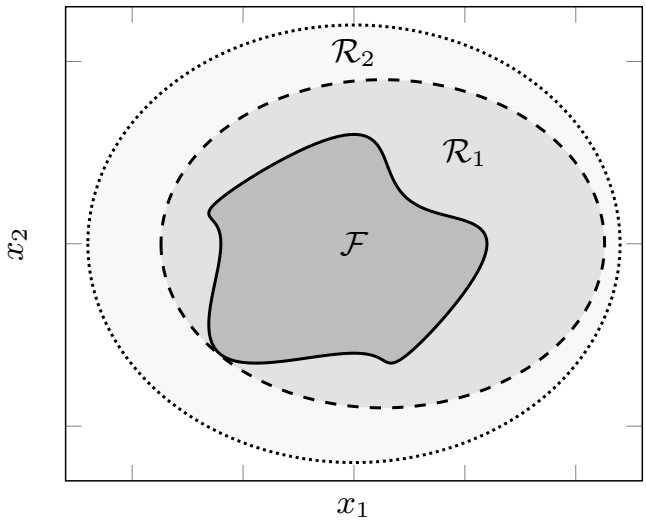


Figure 1.2: Illustration of the feasible sets of a pair of relaxations where the relaxation with $\mathcal{R}_1$ dominates the relaxation with $\mathcal{R}_2$. The sets satisfy $\mathcal{F} \subset \mathcal{R}_1 \subset \mathcal{R}_2$.

1 and 3, the relaxation provides us with a certificate: when the relaxation is exact it is a certificate of global optimality; when the relaxation is infeasible it is a certificate of infeasibility of the original problem. An illustration of an exact relaxation can be seen in Figure 1.5. Suppose that (R) is infeasible. Then (P) is also infeasible, so the relaxation is a certificate of infeasibility. It is generally easier to check infeasibility for a convex optimization problem [63, 11], so it is sensible to use the relaxation to check infeasibility of the original problem.

Different problem classes give rise to different relaxations so there are many relaxations that one can study. In this thesis, our main focus is (subclasses) of *quadratically constrained quadratic programs* (QCQPs) and the so-called *Shor semidefinite programming (SDP) relaxation* which is a semidefinite programming problem. We will interchangeably refer to this relaxation as the Shor relaxation, the semidefinite relaxation, and the SDP relaxation. The details of the SDP relaxation are described in Section 2.1.

The idea of convex relaxation dates back to McCormick in the 1970s [65]. The SDP relaxation was suggested by Shor in the late 1980s [77] and it has proven useful for many hard combinatorial problems [86, 70, 87, 59]. In the past two decades, there has been a surge in proposed relaxations and applications of

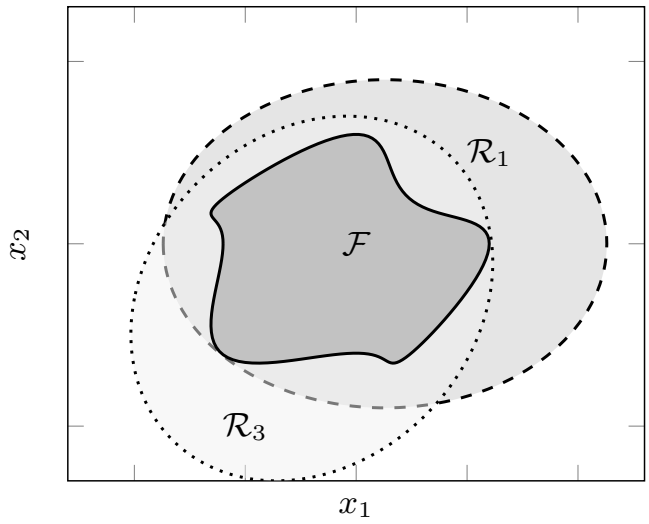


Figure 1.3: Illustration of relaxations that do not dominate each other.

convex relaxation. This can be attributed, in part, to the development of good solvers for convex optimization [69, 41, 79, 82, 91], based on, *e.g.*, interior point methods [2, 71], and modelling software [26, 58], which allows non-experts to easily formulate and solve convex relaxations. Convex relaxation is also a major part of the machinery for solving mixed integer programs; see, *e.g.*, [37, 6, 10].

Convex relaxation techniques should be seen as a complement to local optimization techniques, since it is often paramount, in practice, to find a feasible point no matter how suboptimal it may be. Convex relaxation can help quantify the suboptimality and help inform a decision about whether to look for a new local solution.

Before we describe the challenges that we have tried to address in this thesis, we present an example of a problem that admits an exact semidefinite relaxation to illustrate the technique.

Example 1 *Consider the unconstrained minimization problem*

$$\text{minimize} \quad f(x) = x - x^2 - x^3 + x^4 \tag{1.8}$$

where the variables are $x \in \mathbb{R}$. The objective function can be seen in Figure 1.6. Note that the objective function is nonconvex. Problem (1.8) can be equivalently

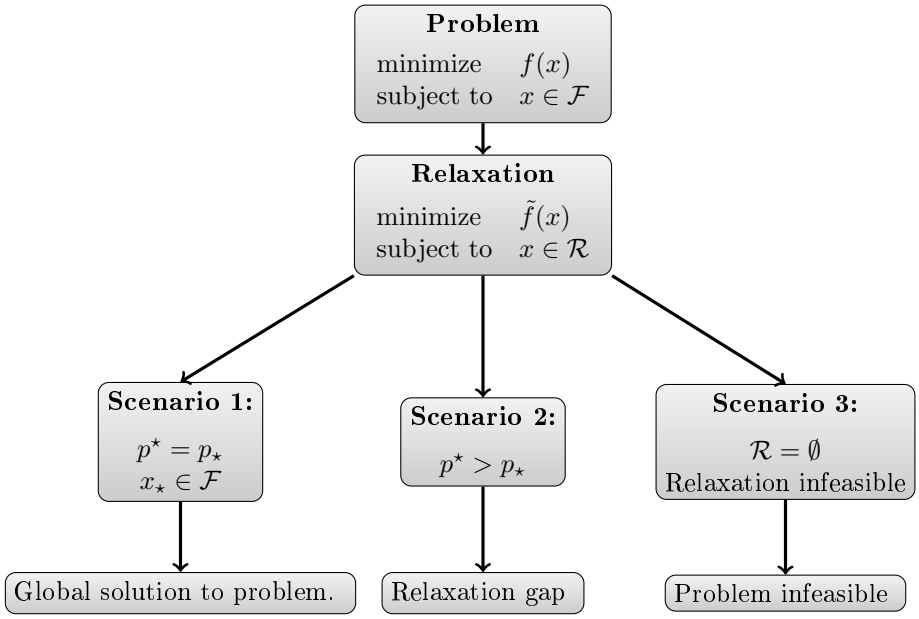


Figure 1.4: Relaxation scenarios.

formulated as the constrained minimization problem

$$\begin{array}{ll} \text{minimize} & \tilde{f}(x,y) = x - y - xy + y^2 \\ \text{subject to} & y = x^2 \end{array} \tag{1.9}$$

where $x, y \in \mathbb{R}$ are the variables. This problem has both a nonconvex objective and a nonconvex constraint, so we have not gained much by this reformulation. The problem can be seen in Figure 1.7 We can equivalently formulate (1.9) as

$$\begin{array}{ll} \text{minimize} & C \bullet Z \\ \text{subject to} & Z_{31} = Z_{22} \\ & Z = \begin{bmatrix} 1 \\ x \\ y \end{bmatrix} \begin{bmatrix} 1 \\ x \\ y \end{bmatrix}^T = \begin{bmatrix} 1 & x & y \\ x & x^2 & xy \\ y & xy & y^2 \end{bmatrix} \end{array} \tag{1.10}$$

where $C = \frac{1}{2}\begin{bmatrix} 0 & 1 & -1 \\ 1 & 0 & -1 \\ -1 & -1 & 2 \end{bmatrix}$. The last constraint satisfies the following equivalence:

$$Z = \begin{bmatrix} 1 \\ x \\ y \end{bmatrix} \begin{bmatrix} 1 \\ x \\ y \end{bmatrix}^T \iff Z \succeq 0 \ \wedge \ \operatorname{rank}(Z) = 1$$

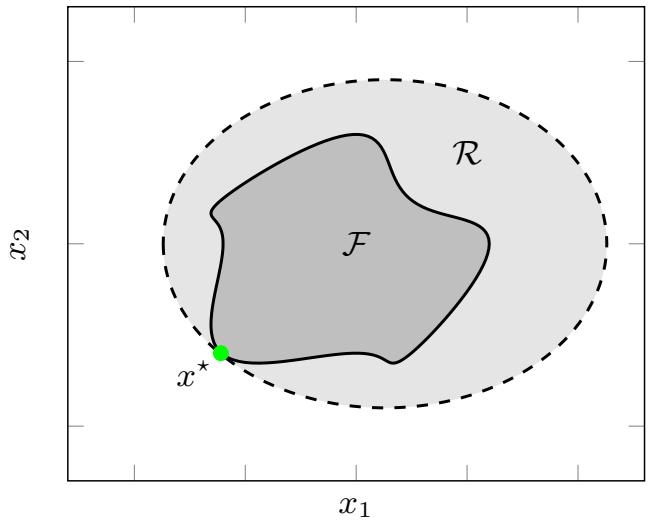


Figure 1.5: Illustration of an exact relaxation. The green circle marks the solution of the relaxation.

where $Z \succeq 0$ denotes that Z must be symmetric and positive semidefinite. So problem (1.8) is equivalent to

$$\begin{array}{ll} \text{minimize} & C \bullet Z \\ \text{subject to} & Z_{31} = Z_{22} \\ & Z_{11} = 1 \\ & Z \succeq 0 \\ & \operatorname{rank}(Z) = 1. \end{array} \tag{1.11}$$

If not for the rank-1 constraint, this would be a convex problem—more precisely a semidefinite program. We obtain an SDP relaxation of (1.8) by dropping the the rank-1 constraint:

$$\begin{array}{ll} \text{minimize} & C \bullet Z \\ \text{subject to} & Z_{31} = Z_{22} \\ & Z_{11} = 1 \\ & Z \succeq 0. \end{array} \tag{1.12}$$

Solving the relaxation (1.12) we obtain the minimizer

$$Z^\star \approx \begin{bmatrix} 1.0000 & -0.6404 & 0.4101 \\ -0.6404 & 0.4101 & -0.2626 \\ 0.4101 & -0.2626 & 0.1682 \end{bmatrix}$$

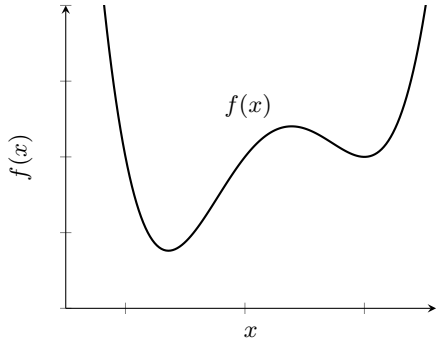


Figure 1.6: Plot of the objective function of (1.8).

with objective value $C \bullet Z^\star \approx -0.6197$. *Checking the eigenvalues, we find that* $\operatorname{rank}(Z^\star) = 1$, *so* $Z^\star$ *is also feasible, and a minimizer, for problem* (1.11). *A rank-1 decomposition of* $Z^\star = z^\star z^{\star T}$ *yields*

$$z^\star \approx \begin{bmatrix} 1 \\ -0.6404 \\ 0.4101 \end{bmatrix}.$$

So the solution (global optimum) to (1.8) *is* $x^\star \approx -0.6404$. *This is plotted in Figure 1.8. Note that* $-z^\star$ *also defines a rank-1 decomposition of* $Z^\star$ *but this can be ruled out since* $z_3^\star$ *must be nonnegative.*

1.2 Challenges

The three papers contained in this thesis address different aspects of convex relaxation techniques, which can be summarized as: scalability, strengthening, and exactness.

As we saw in Example 1, the semidefinite relaxation involves lifting the problem to a higher-dimensional space. For a problem with n variables, the lifting procedure introduces $n(n-1)/2$ new variables and equally many constraints (the rank-1 constraint in (1.10)). The new constraints are then relaxed to a conic inequality. Since the semidefinite relaxation has $O(n^2)$ variables and a conic inequality, there is a concern about the tractability of this relaxation for large scale problems. Therefore, we investigate the scalability of the semidefinite relaxation for the optimal power flow (OPF) problem in Paper A.

When solving a convex relaxation, the hope is that it is exact. Unfortunately,

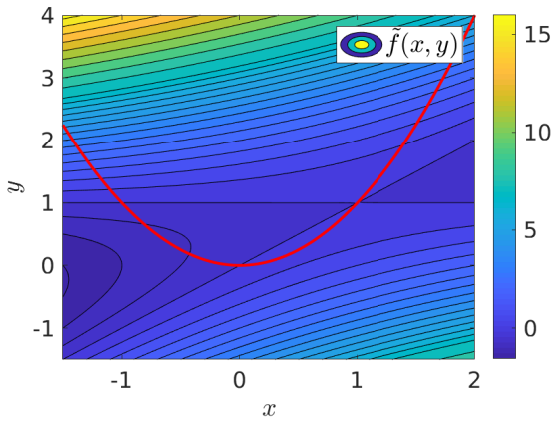


Figure 1.7: Contours and constraint of problem (1.9). The red line is the constraint $y = x^2$.

this is not always the case, which leaves us with different options: Sometimes we can try a different relaxation, sometimes we can strengthen the relaxation, and sometimes there is nothing more to be done. One way to try to strengthen a relaxation is by means of valid inequalities, where constraints of the original feasible set, $\mathcal{F}$, are used to derive constraints for the feasible set $\mathcal{R}$. We explore how this can be done for an extended trust region subproblem in Paper B.

Since exactness is a desired property of a relaxation, it is natural to search for guarantees that the relaxation will be exact. This guarantee can be for a specific relaxation of a specific problem class. In Paper C, We focus on QCQPs with forest structure and present sufficient conditions for exactness of the SDP relaxation based on the data in the problem.

1.3 Notation and Terminology

We introduce the notation and terminology as it becomes relevant through the thesis, but Table 1.1 may be used as a reference. We present some of the notation and concepts here that are central to our presentation.

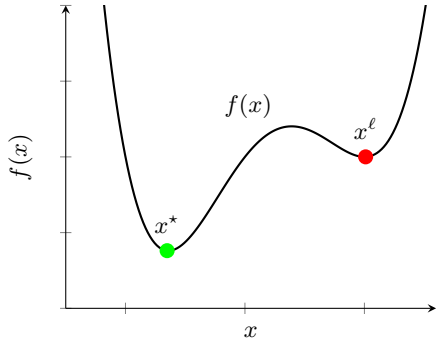


Figure 1.8: Graph of the objective function of (1.8). The green dot marks the global optimum at $x^\star \approx -0.6404$ and the red dot marks a local optimum at $x^\ell \approx 1.0101$.

1.3.1 Convexity

A function, f, is convex if it satisfies

$$\alpha f(x) + (1-\alpha) f(y) \geq f(\alpha x + (1-\alpha) y) \tag{1.13}$$

for all $x, y \in \mathbb{R}^n$, for all $0 \leq \alpha \leq 1$. A set $\mathcal{F}$ is convex if

$$\alpha x + (1-\alpha) y \in \mathcal{F}. \tag{1.14}$$

for all $x, y \in \mathbb{R}^n$, for all $0 \leq \alpha \leq 1$.

1.3.2 Cones

We define the nonnegative orthant as

$$\mathbb{R}^n_+ = \{x \in \mathbb{R}^n : x_i \geq 0,\ i = 1, 2, \ldots, n\}. \tag{1.15}$$

We define the second-order cone (SOC) as

$$\text{SOC} = \{(v_0, v) \in \mathbb{R}^{n+1} : \|v\| \leq v_0\}. \tag{1.16}$$

Let $\mathcal{S}^n \subseteq \mathbb{R}^{n \times n}$ denote the space of symmetric $n \times n$ matrices. We define the cone of symmetric positive semidefinite $n \times n$ matrices as

$$\mathcal{S}^n_+ = \{A \in \mathbb{R}^{n \times n} : A \in \mathcal{S}^n \ \wedge \ x^T A x \geq 0 \ \forall x \in \mathbb{R}^n\}. \tag{1.17}$$

For a cone $\mathcal{K}$, the conic inequality $x \succeq_{\mathcal{K}} 0$ denotes that $x \in \mathcal{K}$; we will omit the subscript $\mathcal{K}$ from $\succeq_{\mathcal{K}}$ when the cone in question is clear from the context. The conic inequality $x \succeq_{\mathcal{K}} y$ means that $x - y \in \mathcal{K}$. For example, for a matrix $A \in \mathcal{S}$, the notation $A \succeq 0$ means that A is positive semidefinite.

1.3.3 Matrices and Graphs

Let A and B be a pair of $n \times m$ matrices. We denote the trace inner product by

$$A \bullet B = \mathrm{tr}(A^T B) = \sum_{i=1}^{n} \sum_{j=1}^{m} A_{ij} B_{ij}$$

An important property of the trace inner product is its cyclic property: for conformable matrices A, B, C we have

$$\mathrm{tr}(ABC) = \mathrm{tr}(BCA). \tag{1.18}$$

For a sparse symmetric $n \times n$ matrix A, we can consider its off-diagonal sparsity pattern as a set of indices, $\mathcal{E}$, indicating the nonzero entries, *i.e.*,

$$(i,j) \in \mathcal{E} \iff i > j \ \wedge \ [A]_{ij} \neq 0. \tag{1.19}$$

Its sparsity graph is then defined as an undirected graph with vertex set $\mathcal{V} = \{1, 2, \ldots, n\}$ and an edge between vertices i and j if $(i,j) \in \mathcal{E}$. For a set of sparse symmetric matrices, $\{A_k\}_{k=0}^{m}$, we define the *aggregate* sparsity pattern and graph as the natural extension of this, *i.e.*, an index (i,j) is contained in $\mathcal{E}$ if *any* of the matrices has a nonzero element:

$$(i,j) \in \mathcal{E} \iff i > j \ \wedge \ \exists k \in \{0, 1, \ldots, n\} : [A_k]_{ij} \neq 0. \tag{1.20}$$

This association between matrix entries and its sparsity graph plays a significant role in Paper C. Diagonal entries correspond to vertices in the graph and off-diagonal entries correspond to edges in the graph. A vertex in a graph is a leaf if it is only connected to one other vertex, *i.e.*, it has only one edge. For a graph, $\mathcal{G}(\mathcal{V}, \mathcal{E})$, we denote the set of leaf vertices by $\mathcal{V}_l$ and the set of diagonal entries associated with leaves by

$$\mathcal{L} = \{(i,i) : i \in \mathcal{V}_l\}. \tag{1.21}$$

We define a non-leaf edge to be an edge where neither of the vertices it connects is a leaf and we denote the set of non-leaf edges by $\mathcal{E}_{\mathrm{nl}}$.

Table 1.1: Overview of notation.

Notation	Description
$\mathbb{R}$	The field of real numbers
$\mathbb{C}$	The field of complex numbers
$\imath$	The imaginary unit ($\imath = \sqrt{-1}$)
$\mathrm{Re}(c)$	The real part, a, of a complex number $c = a + \imath b$.

$\mathrm{Im}(c)$	The imaginary part, b, of a complex number $c = a + \imath b$.
$\mathbb{R}^n_+$	The cone of nonnegative real numbers
SOC	Second-order cone
$\mathcal{S}^n$	The set of $n \times n$ symmetric matrices
$\mathcal{S}^n_+$	The cone of positive semidefinite $n \times n$ symmetric matrices
$\mathcal{H}^n$	The set of $n \times n$ Hermitian matrices
$\mathcal{H}^n_+$	The cone of positive semidefinite $n \times n$ Hermitian matrices
$\mathrm{tr}(A)$	Trace of a $n \times n$ (square) matrix
$A \bullet B$	Inner product of a pair of matrices ($A \bullet B = \sum_{i=1}^{n} \sum_{j=1}^{m} A_{ij} B_{ij}$)
$\mathcal{F}$	Feasible set of original problem
$\mathcal{R}$	Feasible set of relaxation
$\mathcal{G}(\mathcal{V}, \mathcal{E})$	Undirected graph with vertex set $\mathcal{V}$ and edge set $\mathcal{E}$
$\mathcal{V}$	Set vertices ($\mathcal{V} = \{1, 2, \ldots, n\}$)
$\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$	Set edges
$\mathcal{V}_l \subseteq \mathcal{V}$	Set of leaf vertices
$\mathcal{L}$	Entries associated with leaves in the graph
$\mathcal{E}_{\mathrm{nl}}$	Entries associated with non-leaf edges in the graph
e_k	Canonical vector with a 1 in entry k and zeros otherwise
$\overline{E}_{kl} = \frac{1}{2}(e_l e_k^T + e_k e_l^T)$	Matrix with $\frac{1}{2}$ in entries (j,k) and (k,j)
$\widetilde{E}_{kl} = \frac{1}{2_l}(e_l e_k^T + e_k e_l^T)$	Matrix with $\frac{1}{2}$ in entries (j,k) and (k,j)
$\mathcal{O}_n$	The space of orthogonal $n \times n$ matrices

1.4 Outline

In Chapter 2, we describe some common relaxation techniques and the relaxation of some problem classes and applications. In Chapter 3, we discuss exactness and how Paper C contributes to addressing this challenge. In Chapter 4, we discuss how to strengthen a relaxation, when it is not exact, and how Paper B contributes in this area. We also describe a generalization of the valid inequalities suggested in Paper B. In Chapter 5, we describe the alternating current optimal power flow (ACOPF) problem and discuss its impact on this project and vice versa. In Chapter 6, we summarize the contributions of the papers,

which are presented in Appendices A–C, draw conclusions, and discuss future research directions. Appendices D–F contain additional details for the papers and Chapters 3–5.

Chapter 2

Problems and Relaxations

Convex relaxations are often tailored to a specific application or problem class, but many of the techniques are similar. In this chapter, we mainly focus on a specific problem class and a natural relaxation, namely, the class of quadratically constrained quadratic programs (QCQPs) and the semidefinite programming (SDP) relaxation, often called the Shor relaxation. QCQPs are problems where all functions involved—the objective function and the functions in the constraints describing $\mathcal{F}$—are (allowed to be) quadratic. We discuss the semidefinite relaxation of QCQPs in Section 2.1 and, in Section 2.2, we mention some other problem classes and some applications where convex relaxation is often used.

2.1 QCQP and the SDP Relaxation

The relaxation that we have focused on in this project is the SDP relaxation. This is a natural relaxation for QCQPs where both the objective and constraints are quadratic functions. A QCQP can be formulated as

$$\begin{array}{ll} \text{minimize} & x^T A_0 x + a_0^T x \\ \text{subject to} & x^T A_k x + a_k^T x + \alpha_k \leq 0, \quad k = 1, \ldots, m \end{array} \tag{QCQP}$$

where the variables are $x \in \mathbb{R}^n$ and the data are $A_k = A_k^T \in \mathbb{R}^{n\times n}$, $a_k \in \mathbb{R}^n$, and $\alpha_k \in \mathbb{R}$. If all matrices $\{A_k\}_{k=0}^m$ are positive semidefinite, then (QCQP) is a convex problem. Using the trace inner product for conformable matrices and its cyclic property (1.18), the problem (QCQP) is equivalent to

$$\begin{array}{ll} \text{minimize} & \operatorname{tr}(A_0 x x^T) + a_0^T x \\ \text{subject to} & \operatorname{tr}(A_k x x^T) + a_k^T x + \alpha_k \leq 0, \quad k = 1, \ldots, m. \end{array} \tag{2.1}$$

We can now introduce $n(n-1)/2$ new variables, one for each quadratic term (taking symmetry into account), as $X = xx^T$ to have the equivalent problem

$$\begin{array}{ll} \text{minimize} & \operatorname{tr}(A_0 X) + a_0^T x \\ \text{subject to} & \operatorname{tr}(A_k X) + a_k^T x + \alpha_k \leq 0, \quad k = 1, \ldots, m \\ & X = xx^T \end{array} \tag{2.2}$$

where the variables are $x \in \mathbb{R}^n$ and $X \in \mathcal{S}^n$. We call (2.2) the lifted problem, since it is a problem with $n(n-1)/2$ more variables and equally many (quadratic) equality constraint. With this reformulation the objective and the constraints $\operatorname{tr}(A_k X) + a_k^T x + \alpha_k \leq 0$ are linear, while the constraint $X = xx^T$ is nonlinear, since it involves quadratic equality. We can use the fact that

$$X = xx^T \iff X \succeq xx^T \ \wedge \ \operatorname{rank}(X) = 1 \tag{2.3}$$

to obtain the final equivalent problem

$$\begin{array}{ll} \text{minimize} & \operatorname{tr}(A_0 X) + a_0^T x \\ \text{subject to} & \operatorname{tr}(A_k X) + a_k^T x + \alpha_k \leq 0, \quad k = 1, \ldots, m \\ & X \succeq xx^T \\ & \operatorname{rank}(X) = 1. \end{array} \tag{2.4}$$

We can now drop (ignore) the constraint $\operatorname{rank}(X) = 1$ and use that

$$X \succeq xx^T \iff \begin{pmatrix} 1 & x^T \\ x & X \end{pmatrix} \succeq 0 \tag{2.5}$$

to obtain the *Shor semidefinite programming relaxation* [77, 78]:

$$\begin{array}{ll} \text{minimize} & \operatorname{tr}(A_0 X) + a_0^T x \\ \text{subject to} & \operatorname{tr}(A_k X) + a_k^T x + \alpha_k \leq 0, \quad k = 1, \ldots, m \\ & \begin{pmatrix} 1 & x^T \\ x & X \end{pmatrix} \succeq 0. \end{array} \tag{SDR}$$

Loosely speaking, in going from (QCQP) to (SDR), we have unfolded the problem to identify and drop the nonconvexity. It is clear that if a minimizer $(x^\star, X^\star)$ of (SDR) satisfies $X^\star = x^\star (x^\star)^T$, then $(x^\star, X^\star)$ is also a minimizer of (2.4) and, in turn, $x^\star$ is a minimizer of (QCQP).

A different way to arrive at the SDP relaxation is through Lagrangian duality [72, 34]. Consider the Lagrangian for problem (QCQP):

$$L(x,\lambda) = x^T A_0 x + a_0^T x + \sum_{k=1}^{m} \lambda_k (x^T A_k x + a_k^T x + \alpha_k) \tag{2.6}$$

and denote

$$A(\lambda) = A_0 + \sum_{k=1}^{m} \lambda_k A_k,\ a(\lambda) = a_0 + \sum_{k=1}^{m} \lambda_k a_k,\ \alpha(\lambda) = \sum_{k=1}^{m} \lambda_k \alpha_k, \tag{2.7}$$

Then we can formulate the Lagrangian dual problem of (QCQP) as

$$\begin{array}{ll} \text{maximize} & t \\ \text{subject to} & \begin{pmatrix} \alpha(\lambda) - t & \frac{1}{2}a(\lambda)^T \\ \frac{1}{2}a(\lambda) & A(\lambda) \end{pmatrix} \succeq 0 \\ & \lambda \succeq 0. \end{array} \tag{2.8}$$

Problem (2.8) is an SDP and it is also the dual problem of (SDR) [34]. Hence, when there is strong duality between (SDR) and (2.8), these relaxations give the same lower bound. However, for the Lagrangian relaxation the condition for exactness is not as straight-forward as with the SDR.

Consider the feasible set of (2.2) and denote this by $\mathcal{F}_{\text{lifted}}$, *i.e.*,

$$\mathcal{F}_{\text{lifted}} = \left\{ (x,X) \in \mathbb{R}^n \times \mathcal{S}^n : \begin{array}{l} \operatorname{tr}(A_k X) + a_k^T x + \alpha_k \le 0, \quad k = 1,\dots,m \\ X = xx^T \end{array} \right\}. \tag{2.9}$$

From a relaxation perspective we are interested in

$$\mathcal{C}_{\text{lifted}} = \overline{\text{conv}}\left\{(x,X) : (x,X) \in \mathcal{F}_{\text{lifted}}\right\}. \tag{2.10}$$

Since the objective function is linear in (x,X), having a tractable representation of $\mathcal{C}_{\text{lifted}}$ [21] would allow us to solve (QCQP) by solving

$$\min_{x,X}\{A_0 \bullet X + a_0^T x : (x,X) \in \mathcal{C}_{\text{lifted}}\}. \tag{2.11}$$

However, we only have tractable representations of $\mathcal{C}_{\text{lifted}}$ in some special cases [19].

There are other ways to obtain a relaxation of (QCQP) and the problem is often augmented with additional constraints, whose structure can then be exploited [3, 50]. For a survey and comparison of other relaxations, see, *e.g.*, [8].

2.1.1 Complex-Valued QCQP

A complex-valued QCQP is very similar to a real-valued QCQP (QCQP), except the variables are complex and the matrices are Hermitian. A complex-valued QCQP can be formulated as

$$\begin{array}{ll} \text{minimize} & z^H Q_0 z + \operatorname{Re}(q_0^H z) \\ \text{subject to} & z^H Q_k z + \operatorname{Re}(q_k^H z) + \phi_k \leq 0, \quad k = 1, \ldots, m \end{array} \tag{C-QCQP}$$

where $z \in \mathbb{C}^n$ are the variables and the data is $Q_k = Q_k^H \in \mathbb{C}^{n \times n}$, $q_k \in \mathbb{C}^n$, $\phi_k \in \mathbb{R}$. This problem formulation has many applications in signal processing [64, 45, 23] and is also a natural formulation for the OPF problem with some assumptions.

Any complex-valued QCQP (C-QCQP) can be mapped to an equivalent real-valued QCQP. Define the matrices

$$T_k = \begin{pmatrix} \operatorname{Re}(Q_k) & -\operatorname{Im}(Q_k) \\ \operatorname{Im}(Q_k) & \operatorname{Re}(Q_k) \end{pmatrix} \tag{2.12}$$

and the vectors

$$t_k = \begin{pmatrix} \operatorname{Re}(q_k) \\ \operatorname{Im}(q_k) \end{pmatrix} \tag{2.13}$$

Then (C-QCQP) can be expressed as

$$\begin{array}{ll} \text{minimize} & \binom{x}{y}^T T_0 \binom{x}{y} + t_0^T \binom{x}{y} \\ \text{subject to} & \binom{x}{y}^T T_k \binom{x}{y} + t_k^T \binom{x}{y} + \phi_k \leq 0, \quad k = 1, \ldots, m \end{array} \tag{2.14}$$

where the variables are $x, y \in \mathbb{R}^n$ and the data is $T_k \in \mathbb{R}^{2n \times 2n}$, $t_k \in \mathbb{R}^{2n}$, $\phi_k \in \mathbb{R}$.

Since any complex-valued QCQP can be mapped to an equivalent real-valued QCQP, it might seem odd to even consider the complex-valued QCQP at all. However, since the converse is not true (any real-valued QCQP does not have an equivalent formulation as a complex-valued QCQP), there is some structure that may be exploited. For example, there are dedicated exactness results [9, 46] and strengthening results [51] for relaxations of (C-QCQP).

For complex-valued problems, there is a choice of representing the variables in rectangular or polar coordinates and this choice can lead to different relaxations. In Chapter 5, we consider the OPF problem in rectangular variables.

2.2 Other Problems and Applications

Combinatorial optimization problems often include binary or integer variables. Hence, we are no longer in the realm of continuous optimization, but we can use the QCQP problem as a bridge between combinatorial optimization and continuous optimization, since a binary constraint $x \in \{0, 1\}$ can be expressed as the equivalent quadratic constraint $x(1-x) = 0$ and relaxed by the lifting described in Section 2.1, or relaxed to the linear constraint $0 \leq x \leq 1$. There are other relaxations for problems with binary variables, so-called (0,1)-programs, and these problems have received particular attention in the literature [60, 74, 43, 53]. Optimization of binary nonlinear programs is closely related to the study of pseudo-Boolean optimization [16]. For more details on relaxation of combinatorial problems and some applications of SDP relaxations, see, *e.g.*, [42, 84, 87]. Convex relaxations are also widely used in branch-and-bound algorithms, where it is important to obtain good lower bounds on the optimal value [10].

For polynomial problems, *i.e.*, problems where $f, g_1, \ldots, g_m$ in (P) are polynomials, there exists a hierarchy of SDP relaxations of growing size whose solutions converge to the solution of the original problem [52]. This is based on sum of squares polynomials and the dual theory of the moment relaxation. The interested reader is referred to [54].

As mentioned in the introduction in Chapter 1, the development of good solvers and modelling tools has spurred an increase in the use of convex relaxation techniques for various applications. Here, we provide a short list of some applications:

- Graph applications: MaxCut [39, 75], community detection [1], synchronization [7], offset selection for traffic signals [25].
- Signal processing [64]: Multiple input multiple output (MIMO) [66], phase recovery [85].
- Power systems: optimal power flow (OPF) [68]. This application has played a large role in this thesis and is the subject of Chapter 5.

One of the most well-known results of convex relaxation is for the Max-Cut problem, where the goal is to find a two-way partitioning of a weighted graph such that the edges connecting the two sets have maximum weight. Goemans and Williamson [39] presented a randomized algorithm, based on SDP relaxation, guaranteed to deliver a feasible point with objective value at least 0.87856 times the optimal solution.

Chapter 3

Exactness

Out of the three scenarios depicted in Figure 1.4, scenario 1 is arguably the best. In other words, a much desired property of a convex relaxation is *exactness*. When a problem has an exact efficiently solvable convex relaxation, we can solve the original problem in polynomial time. This does not mean that the original problem is convex; its solution just happens to coincide with that of a convex problem. It is non-trivial to pin-point when this will happen. Hence, the exactness of a relaxation is often gauged on a solve-and-check basis, *i.e.*, the relaxation is solved and some criterion for exactness is checked.

In this chapter, we consider exactness guarantees, *i.e.*, a theoretical certificate that a specific relaxation will provide a solution for an instance of a specific problem. It is unrealistic to expect such a guarantee for a class of NP-hard problems with a polynomial-time convex relaxation. Hence, we need to restrict our considerations to problems where the objective function and feasible set have a specific structure. For example, it is well-known that the trust region subproblem (TRS) can be solved in polynomial time, see, *e.g.*, [88, 80], since it has an exact relaxation. This problem takes the form

$$\begin{array}{ll} \text{minimize} & x^T H x + g^T x \\ \text{subject to} & \|x\| \leq 1 \end{array} \tag{3.1}$$

where the variables are $x \in \mathbb{R}^n$ and the data is $H = H^T \in \mathbb{R}^{n \times n}$ and $g \in \mathbb{R}^n$. This is a problem with a (nonconvex) quadratic objective where $\mathcal{F}$ is the unit

Euclidean ball. Other problems that have a relaxation that admits no relaxation gap can be found in, *e.g.*, [89, 9, 19, 12].

The condition that we propose in paper C does not guarantee exactness for a full class of problems. Instead, we present conditions that can be checked for an *instance* of a specific problem class, *i.e.*, for a problem where the data is given. We consider the SDP relaxation of QCQPs of the form

$$\begin{array}{ll} \text{minimize} & x^H A_0 x \\ \text{subject to} & x^H A_k x + \alpha_k \leq 0, \quad k = 1, \ldots, m, \end{array} \tag{T-QCQP}$$

where the variables are $x \in \mathbb{C}^n$, the data are $\{A_k\}_{k=0}^m$ and $\{\alpha_k\}_{k=1}^m$, and where the aggregate sparsity graph of the matrices $\{A_k\}_{k=0}^m$ is a forest. The SDP relaxation is:

$$\begin{array}{ll} \text{minimize} & A_0 \bullet X \\ \text{subject to} & A_k \bullet X + \alpha_k \leq 0, \quad k = 1, \ldots, m \\ & X \succeq 0 \end{array} \tag{3.2}$$

where the variables are $X = X^H \in \mathbb{C}^{n \times n}$. The dual of the SDP relaxation (and of (T-QCQP)) is:

$$\begin{array}{ll} \text{maximize} & \alpha^T \lambda \\ \text{subject to} & Y = A_0 + \sum_{k=1}^m \lambda_k A_k \\ & Y \succeq 0 \\ & \lambda \succeq 0 \end{array} \tag{3.3}$$

where the variables are $\lambda \in \mathbb{R}^m$ and $Y = Y^H \in \mathbb{C}^{n \times n}$ and α is a column vector with entries α_k. Note that we could eliminate the variable Y (and have the linear matrix inequality $A_0 + \sum_{k=1}^m \lambda_k A_k \succeq 0$), but it is convenient to keep it around, since it is the dual variable for $X \succeq 0$ and it plays an important role in the exactness condition. We denote the feasible set of the dual problem (3.3) by

$$\Omega = \left\{ (\lambda, Y) : \begin{array}{l} Y = A_0 + \sum_{k=1}^m \lambda_k A_k \\ Y \succeq 0 \\ \lambda \succeq 0 \end{array} \right\}. \tag{3.4}$$

A general QCQP of the form (QCQP) can be formulated as the equivalent homogeneous QCQP

$$\begin{array}{ll} \text{minimize} & \begin{pmatrix} 1 \\ x \end{pmatrix}^T \begin{pmatrix} 0 & \frac{1}{2} a_0^T \\ \frac{1}{2} a_0 & A_0 \end{pmatrix} \begin{pmatrix} 1 \\ x \end{pmatrix} \\ \text{subject to} & \begin{pmatrix} 1 \\ x \end{pmatrix}^T \begin{pmatrix} \alpha_k & \frac{1}{2} a_k^T \\ \frac{1}{2} a_k & A_k \end{pmatrix} \begin{pmatrix} 1 \\ x \end{pmatrix} \leq 0, \quad k = 1, \ldots, m. \end{array} \tag{3.5}$$

Hence, we can apply the theory of paper C to a general QCQP if the aggregate sparsity graph of the matrices

$$\left\{ \begin{pmatrix} \alpha_k & \frac{1}{2}a_k^T \\ \frac{1}{2}a_k & A_k \end{pmatrix} \right\}_{k=0}^{m}$$

is a forest.

In the following we summarize paper C. In the paper we have a numerical example to illustrate the conditions developed; after the summary in Section 3.1, we give some additional examples in Section 3.2.

3.1 Summary of Paper C

In this section, we summarize and discuss the sufficient condition for exactness of the SDP relaxation of an instance of (T-QCQP) presented in Paper C in Appendix C. To ease the discussion, we will refer to the data $\{A_k\}_{k=0}^m$ as *the data matrices* and the data $\{\alpha_k\}_{k=1}^m$ as *the data vector*.

One way to guarantee exactness of the semidefinite relaxation is to guarantee that a solution $X^\star$ is rank-1 as we saw in Section 2.1. The condition in paper C relies on strong duality between (3.2) and (3.3) and the result that a positive semidefinite matrix whose sparsity graph is a connected tree has rank at least $n-1$ [49] and an extension of this result. Let $X^\star$ be a solution to (3.2) and let $(\lambda^\star, Y^\star)$ be a solution to (3.3). From strong duality and complementary slackness [72] we have $X^\star \bullet Y^\star = 0$, which implies that $X^\star Y^\star = 0$, since both matrices are positive semidefinite by their feasibility. Sylvester's inequality (see, *e.g.*, [44]) gives us the bound:

$$\operatorname{rank} X^\star + \operatorname{rank} Y^\star \leq n + \operatorname{rank}(X^\star Y^\star) = n. \tag{3.6}$$

Hence, if we can guarantee that $\operatorname{rank}(Y^\star) \geq n-1$, then we have that $\operatorname{rank}(X^\star) \leq 1$ and the relaxation is exact. This is the basis of the condition of paper C and we need the tree structure of Y to utilize a result about the multiplicity of the smallest eigenvalue for matrices with connected tree structure. We use a number of feasibility systems that check if it is possible to introduce a zero in strategic entries of $Y^\star$. We call these the *essential* feasibility systems and the indices of these entries comprise the set $\mathcal{E}_{\text{ess}}$. In particular, the essential feasibility systems are those associated with leaves and non-leaf edges in the sparsity graph, so $\mathcal{E}_{\text{ess}} = \mathcal{L} \cup \mathcal{E}_{\text{nl}}$. Here, $\mathcal{L}$ is the set of diagonal entries corresponding to a leaf in the aggregate sparsity graph and $\mathcal{E}_{\text{nl}}$ is the set of entries corresponding to non-leaf edges in the aggregate sparsity graph.

Let $\mathbf{O}_{ij} = \mathbb{R}^m \times \{Y \in \mathcal{H}^n : Y_{ij} = Y_{ji} = 0\}$ denote the direct product of R^m and space of Hermitian matrices with a zero in entry (i, j). Then we can express a feasibility system as checking if the set

$$\Omega \cap \mathbf{O}_{ij} \tag{3.7}$$

is empty, *i.e.*, checking if there exists a dual feasible pair (λ, Y) for which $Y_{ij} = 0$. Then the problem has an exact relaxation if

$$\bigcup_{(i,j)\in\mathcal{E}_{\text{ess}}} (\Omega \cap \mathbf{O}_{ij}) = \emptyset. \tag{3.8}$$

This checks if there exists a feasible pair (λ, Y) of (3.3) for which the sparsity graph of Y does *not* have the desired structure (connected tree or a variant thereof). Each feasibility system $\Omega \cap \mathbf{O}_{ij}$ can be formulated as an SOCP, due to the presence of tree structure [83]; this derivation can be found in Appendix D.1.

When the condition (3.8) holds for an instance (a problem with given data $\{\hat{A}_k\}_{k=0}^m$ and $\{\hat{\alpha}_k\}_{k=1}^m$), we refer to this as the *nominal* instance. The condition depends only on the data matrices, so if it holds, an instance of (T-QCQP) with those nominal data matrices and *any* data vector has an exact relaxation, given that the instance is feasible.

Given a nominal instance, *i.e.*, one where (3.8) holds, we propose to use perturbation sets to extend the exactness guarantee to instances for which the data matrices are close to the nominal data matrices in a certain sense. These perturbation sets can be chosen to reflect any known uncertainty in the data. We use these perturbation sets to formulate a set of *robust* feasibility systems. These can be used to find a region of exactness, *i.e.*, a set of perturbations of the nominal data matrices for which the resulting instance is guaranteed to have an exact relaxation.

When one of the essential feasibility systems is feasible, the condition (3.8) does not hold, and we can not guarantee exactness for that instance. However, the instance may still have an exact relaxation and a solution $Y^\star$ to (3.3) which has the desired sparsity graph. In this case, we propose a number of *restricted* feasibility systems with the aim of guaranteeing exactness for instances with the same data matrices and data vectors that are close to the nominal data vectors.

Given an instance with data $\{\hat{A}_k\}_{k=0}^m$ and $\{\hat{\alpha}_k\}_{k=1}^m$ the process of checking exactness of its relaxation is demonstrated in Figure 3.1. The figure also illustrates when the robust and restricted feasibility systems become relevant.

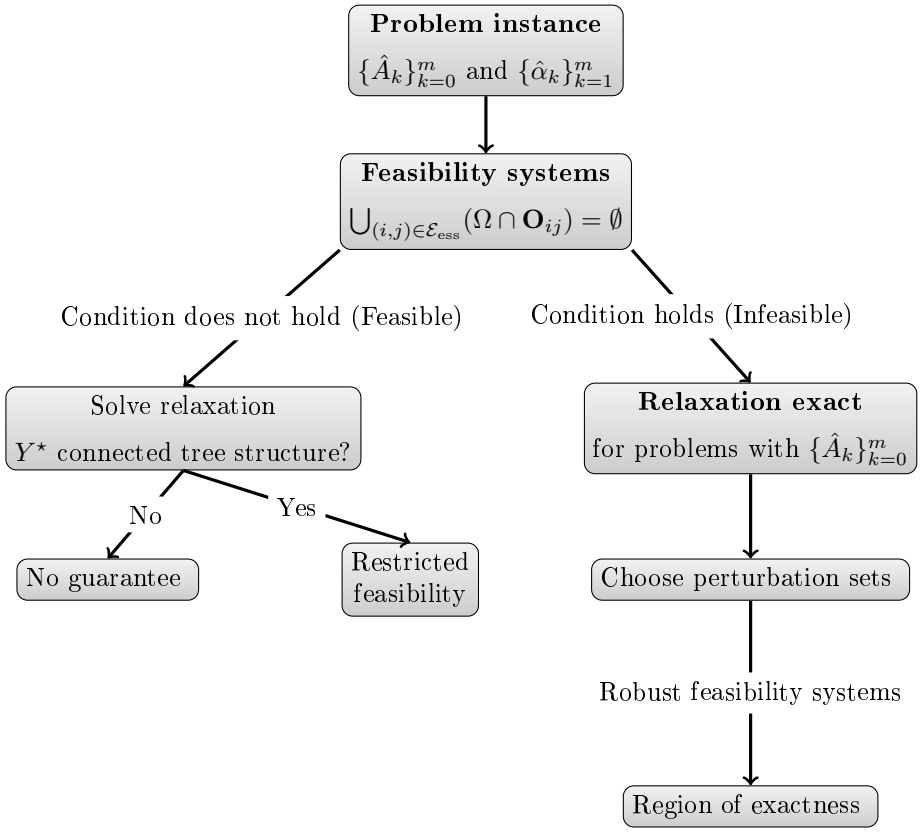


Figure 3.1: Process of checking exactness of the SDP relaxation for an instance of (T-QCQP).

3.2 Additional Examples

In this section, we give some examples of the application of the sufficient condition (3.8).

3.2.1 Two-Dimensional Problem

Consider a two-dimensional ($x \in \mathbb{R}^2$) general QCQP:

$$\begin{array}{ll} \text{minimize} & c_{11}x_1^2 + c_{22}x_2^2 + 2\,c_{12}x_1x_2 + 2\,c_1x_1 + 2\,c_2x_2 \\ \text{subject to} & a_{k,11}x_1^2 + a_{k,22}x_2^2 + 2\,a_{k,12}x_1x_2 + 2\,a_{k,1}x_1 + 2\,a_{k,2}x_2 + \gamma_k \leq 0, \\ & \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad k = 1, \ldots, m. \end{array} \tag{3.9}$$

where the variables are $x_1, x_2 \in \mathbb{R}$ and the data are $c_1, c_2, c_{11}, c_{12}, c_{22} \in \mathbb{R}$ and $a_{k,1}, a_{k,2}, a_{k,11}, a_{k,12}, a_{k,22}, \gamma_k \in \mathbb{R}$ $(k = 1, \ldots, m)$. This problem can be formulated as a homogeneous problem of the form

$$\begin{array}{ll} \text{minimize} & \begin{pmatrix} 1 \\ x_1 \\ x_2 \end{pmatrix}^T \begin{pmatrix} 0 & c_1 & c_2 \\ c_1 & c_{11} & c_{12} \\ c_2 & c_{12} & c_{22} \end{pmatrix} \begin{pmatrix} 1 \\ x_1 \\ x_2 \end{pmatrix} \\ \text{subject to} & \begin{pmatrix} 1 \\ x_1 \\ x_2 \end{pmatrix}^T \begin{pmatrix} \gamma_k & a_{k,1} & a_{k,2} \\ a_{k,1} & a_{k,11} & a_{k,12} \\ a_{k,2} & a_{k,12} & a_{k,22} \end{pmatrix} \begin{pmatrix} 1 \\ x_1 \\ x_2 \end{pmatrix} \leq 0, \quad k = 1, \ldots, m. \end{array} \tag{3.10}$$

The matrices of this problem will have an aggregate sparsity pattern that is a forest, if there are no bilinear terms in the problem ($c_{12} = a_{1,12} = \cdots = a_{m,12} = 0$) or if either of the linear terms is not present ($c_1 = a_{1,1} = \cdots = a_{m,1} = 0$ or $c_2 = a_{1,2} = \cdots = a_{m,2} = 0$). When this is the case, we can apply the condition (3.8) to check if the problem will have an exact SDP relaxation. In the following example, we go through the check of the exactness condition (3.8) for a two-dimensional problem.

Example 3.1 *Consider the problem*

$$\begin{array}{ll} \text{minimize} & x^T D x + 2\,g^T x \\ \text{subject to} & x_1^2 + x_2^2 \leq 1 \\ & x_1 \geq 0 \\ & x_2 \geq 0 \end{array} \tag{3.11}$$

where the variables are $x \in \mathbb{R}^2$ and the data is

$$D = \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix}, \; g = \begin{pmatrix} -2 \\ 1 \end{pmatrix}.$$

The feasible set and contours of the problem can be seen in Figure 3.2. We can

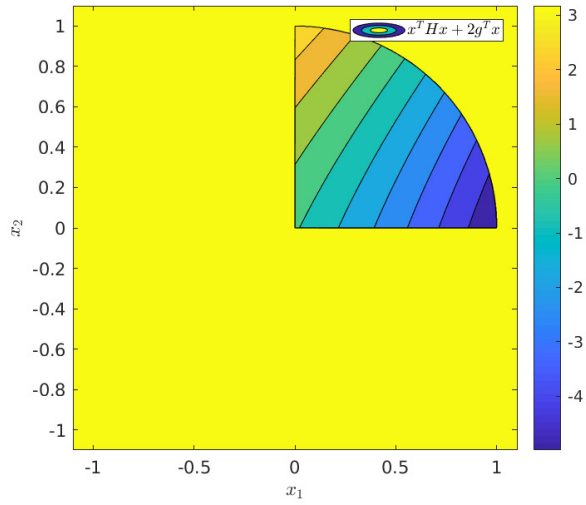


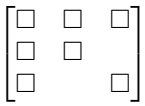
Figure 3.2: Feasible set and contours of problem (3.11). Yellow indicates infeasible points.

formulate this in the form of (T-QCQP) *with* $\alpha = (-1, 0, 0, -1, 1)$ *and*

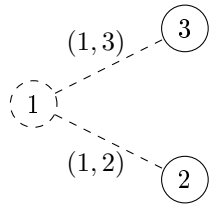
$$A_0 = \begin{pmatrix} 0 & -2 & 1 \\ -2 & -1 & 0 \\ 1 & 0 & 1 \end{pmatrix}, A_1 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, A_2 = \frac{1}{2}\begin{pmatrix} 0 & -1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix},$$
$$A_3 = \frac{1}{2}\begin{pmatrix} 0 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{pmatrix}, A_4 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, A_5 = \begin{pmatrix} -1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}.$$

The aggregate sparsity pattern of these matrices can be seen in Figure 3.3. The essential feasibility systems are the indices $(2,2)$ *and* $(3,3)$*. Hence, to check the exactness condition* (3.8)*, we need to check if there exists* $(\lambda, Y) \in \Omega$ *such that* $Y_{22} = 0$ *or* $Y_{33} = 0$*. We can express the* $(2,2)$ *feasibility system as checking if there exists* $\lambda \in \mathbb{R}_5$ *such that*

$$\begin{pmatrix} 0 & -2 & 1 \\ -2 & -1 & 0 \\ 1 & 0 & 1 \end{pmatrix} + \lambda_1 \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} + \lambda_2 \frac{1}{2}\begin{pmatrix} 0 & -1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} +$$
$$\lambda_3 \frac{1}{2}\begin{pmatrix} 0 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{pmatrix} + \lambda_4 \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} + \lambda_5 \begin{pmatrix} -1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \succeq 0 \qquad (3.12)$$
$$-1 + \lambda_1 = 0$$
$$\lambda \succeq 0.$$



(a) Aggregate sparsity pattern of the matrices in problem (3.11). The symbol □ denotes a nonzero and the lack of symbol denotes a zero.



(b) Sparsity graph: the vertices correspond to diagonal entries and edges correspond to off-diagonal entries. The solid lines correspond to the essential feasibility systems.

Figure 3.3: Aggregate sparsity pattern and aggregate sparsity graph of problem (3.11).

By manual inspection we can see that this is infeasible since a zero on the diagonal of a positive semidefinite matrix implies zeros in that row and column but entry $(2,1)$ *of the linear matrix inequality (LMI) reads* $-2-\lambda_2=0$ *which cannot be zero since* $\lambda_2 \geq 0$.

Similarly, we can express the $(3,3)$ *feasibility system as checking if there exists* $\lambda \in \mathbb{R}_5$ *such that*

$$\begin{aligned} &\begin{pmatrix} 0 & -2 & 1 \\ -2 & -1 & 0 \\ 1 & 0 & 1 \end{pmatrix} + \lambda_1 \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} + \lambda_2 \frac{1}{2} \begin{pmatrix} 0 & -1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} + \\ &\lambda_3 \frac{1}{2} \begin{pmatrix} 0 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{pmatrix} + \lambda_4 \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} + \lambda_5 \begin{pmatrix} -1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \succeq 0 \\ &1 + \lambda_1 = 0 \\ &\lambda \succeq 0. \end{aligned} \tag{3.13}$$

This is clearly infeasible since $\lambda_1 \geq 0$.

Hence, both essential feasibility systems are infeasible, so (3.8) *holds and we can guarantee that the semidefinite relaxation of* (3.11) *is exact for any* α *for which the problem is feasible.*

3.2.2 Example in the Complex Domain

An exactness condition that is related to (3.8) is given in [17] and discussed in Paper C. The condition can be stated as: the relaxation is exact if $0 \notin \text{int conv}\{[A_0]_{ij}, [A_1]_{ij}, \ldots, [A_m]_{ij}\}$ for all $(i,j) \in \mathcal{E}$. That is, if zero is not in the interior of the convex hull of the point set defined by the off-diagonal entries (in the complex plane) for all off-diagonal entries, then the relaxation is exact. In the following example, this does not hold, but the condition (3.8) does hold.

We remark that, due to the bound $r^\star \leq \lfloor \sqrt{m} \rfloor = \lfloor \sqrt{3} \rfloor = 1$ [46, 57], the existence of a rank-1 solution is already guaranteed.

Example 2 *Consider the complex QCQP*

$$\begin{array}{ll} \text{minimize} & x^H C_0 x \\ \text{subject to} & x^H C_k x + b_k \leq 0, \quad k = 1, \ldots, 3, \end{array} \tag{3.14}$$

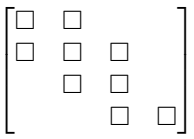
where the variables are $x \in \mathbb{C}^4$ *and the data are* $b_1 = 0, b_2 = 0, b_3 = 1$ *and*

$$C_0 = \begin{pmatrix} 4 & 1-i & 0 & 0 \\ 1+i & 4 & 2+i & 0 \\ 0 & 2-i & 4 & -1+i \\ 0 & 0 & -1-i & 4 \end{pmatrix}, C_1 = \begin{pmatrix} -3 & 1+2i & 0 & 0 \\ 1-2i & 0 & -1-\frac{i}{2} & 0 \\ 0 & -1+\frac{i}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix},$$

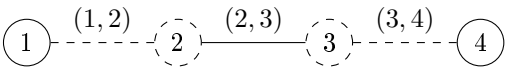
$$C_2 = \begin{pmatrix} 0 & 1-i & 0 & 0 \\ 1+i & 0 & 0 & 0 \\ 0 & 0 & 0 & 1+i \\ 0 & 0 & 1-i & 0 \end{pmatrix}, C_3 = \begin{pmatrix} 0 & -1-i & 0 & 0 \\ -1+i & 0 & 0 & 0 \\ 0 & 0 & 0 & -1-i \\ 0 & 0 & -1+i & 0 \end{pmatrix}.$$

All matrices are Hermitian, C_0 *is positive definite, and* C_1*,* C_2*, and* C_3 *are indefinite. The aggregate sparsity pattern and aggregate sparsity graph are shown in Figure 3.4. From Figure 3.5, we see that this problem does not satisfy the condition in [17], since the convex hull of* $\{[C_0]_{34}, [C_1]_{34}, [C_2]_{34}, [C_3]_{34}\}$ *contains zero in its interior.*

For this problem the essential feasibility systems are $\{(1,1), (2,3), (4,4)\}$ *and they are all infeasible, so we can guarantee that this problem has an exact relaxation regardless of the data* b_1, b_2, b_3*, given that the problem is feasible.*

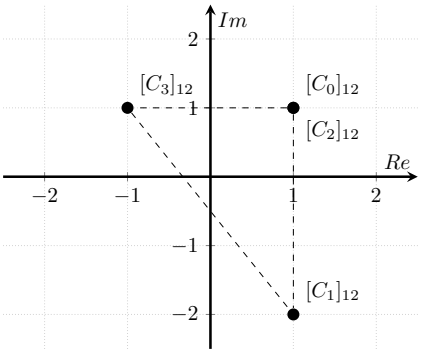


(a) Aggregate sparsity pattern of the matrices in problem (3.14). The symbol □ denotes a nonzero and the lack of symbol denotes a zero.

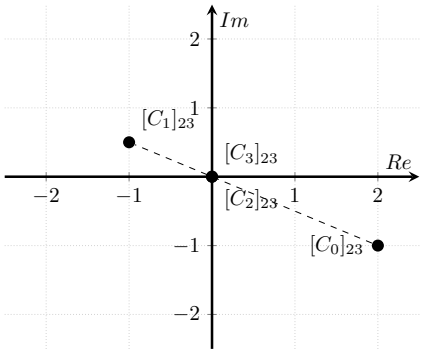


(b) Sparsity graph: the vertices correspond to diagonal entries and edges correspond to off-diagonal entries. The solid lines correspond to the essential feasibility systems.

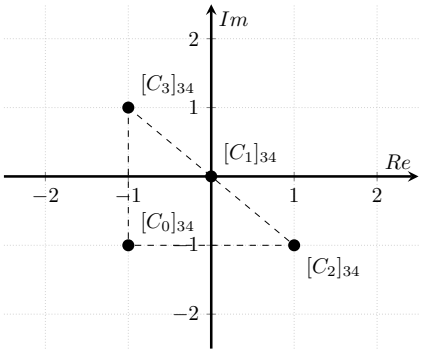
Figure 3.4: Aggregate sparsity pattern and aggregate sparsity graph of problem (3.14).



(a) (2,1)



(b) (3,2)



(c) (3,4)

Figure 3.5: Off-diagonal point sets in the complex plane. The dashed lines sketch the convex hull.

Chapter 4

Strengthening

In this chapter we consider strengthening techniques. These can be used to obtain a stronger relaxation and can help us in our pursuit of exactness. When we relax a problem, we usually throw some information away in order to obtain a problem that we can solve. In the case of convex relaxation, we disregard nonconvexities to obtain a convex problem. Depending on the formulation of the problem, different pieces of information might be thrown away by different relaxations. In this chapter, we consider *valid inequalities*. The idea of valid inequalities is to squeeze some more information out of the original problem—information that can be used in the relaxation. With this additional information, we may be able to obtain a stronger relaxation.

Suppose that we have a problem with feasible set

$$\mathcal{F} = \{x \in \mathbb{R}^n : g_i(x) \leq 0,\ i = 1, \ldots, m\}, \tag{4.1}$$

and a relaxation with feasible set $\mathcal{R}$. One idea for obtaining a stronger relaxation is to look for a set K such that

$$\mathcal{F} \subseteq \mathcal{R} \cap K \subset \mathcal{R}. \tag{4.2}$$

We call the set K a *cut*, since it "cuts away" some of the feasible set of the relaxation. Note that $\mathcal{F} \subseteq K$, so the cut has to contain the feasible set of the problem. The situation is illustrated in Figure 4.1.

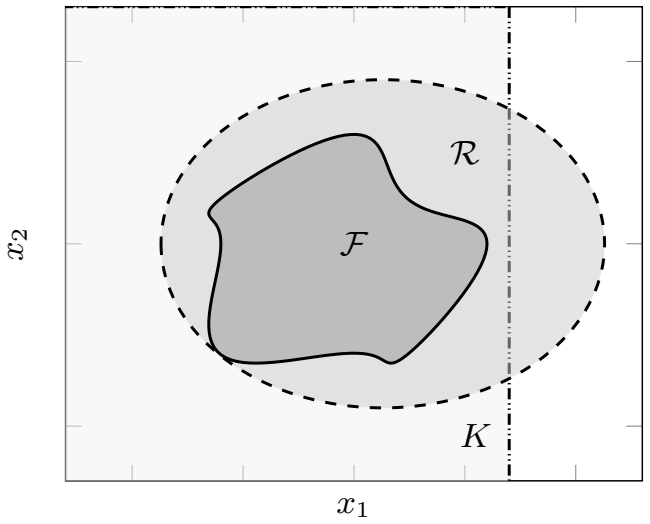


Figure 4.1: Illustration of a cut K.

One way to derive cuts is to use valid inequalities. A valid inequality for a feasible set is a constraint $h(x) \geq 0$ such that

$$x \in \mathcal{F} \implies h(x) \geq 0. \tag{4.3}$$

Hence, adding the constraint $h(x) \geq 0$ to the problem does not change the problem. Therefore, a valid inequality $h(x) \geq 0$ is also called a redundant constraint, since it is implied by the other constraints. However, to obtain the SDP relaxation, the original problem is lifted and relaxed, so the valid inequality may not be redundant for the feasible set of the relaxation. As a consequence, we can add the valid inequality, or cut, to the relaxation and perhaps obtain a stronger relaxation.

A valid inequality can be obtained by combining the constraints of $\mathcal{F}$ in some way. Recall the relationship in Section 2.1 between the feasible set $\mathcal{F}$ and the lifted feasible set $\mathcal{F}_{\text{lifted}}$. In particular, note that a constraint $g_i(x) \geq 0$ that is quadratic in x is linear in (x, X) in the lifted feasible set. As a consequence, any valid quadratic constraint $h(x) \geq 0$ for $\mathcal{F}$ can be added to the SDP relaxation. In paper B, we derive a class of valid quadratic inequalities for a specific $\mathcal{F}$. Before we describe these and summarize paper B, we mention three techniques that can be used for obtaining valid inequalities. There exist different techniques that can be used for specific feasible sets; see, *e.g.*, [48] for techniques for QCQP, or [43, 28, 29] for techniques in combinatorial optimization. We focus on three

techniques which generally combine a pair of constraints in some way. For example the reformulation-linearization technique (RLT) combines a pair of linear constraints to obtain a valid quadratic inequality:

$$v_1^T x + a_1 \geq 0 \wedge v_2^T x + a_2 \geq 0 \implies x^T v_1 v_2^T x + (a_1 v_2^T + a_2 v_1^T) x + a_1 a_2 \geq 0. \quad (4.4)$$

We summarize the techniques to obtain RLT constraints, SOCRLT constraint, and Kronecker SOC constraints in Table 4.1 by the type of constraints they combine. Note that a SOC constraint can be expressed as an SDP constraint:

$$\|x\| \leq R \iff \begin{pmatrix} R & x^T \\ x & RI \end{pmatrix} \succeq 0 \quad (4.5)$$

where I denotes the identity matrix. Hence, a pair of SOC constraints can be used to obtain a valid KSOC constraint.

Table 4.1: Techniques for obtaining valid inequalities. Here, the symbol $\times$ denotes multiplication and the symbol $\otimes$ denotes the Kronecker product.

Constraints	Types of constraints combined
RLT[76]	Linear $\times$ Linear
SOCRLT[89, 20]	Linear $\times$ SOC
KSOC[4]	SDP/SOC $\otimes$ SDP/SOC

4.1 Summary of Paper B

In paper B, we add to the techniques for strengthening relaxations by deriving a new class of valid inequalities. We derive the inequalities for an extended trust region subproblem of the form

$$\begin{aligned} \text{minimize} \quad & x^T H x + 2\, g^T x && (4.6a) \\ \text{subject to} \quad & r \leq \|x\| \leq R && (4.6b) \\ & \|x - c\| \leq b^T x - a, && (4.6c) \end{aligned}$$

where the variables are $x \in \mathbb{R}^n$ and the data are $H = H^T \in \mathbb{R}^{n \times n}$, $g, c, b \in \mathbb{R}^n$, $\alpha \in \mathbb{R}$, and $r, R \in \mathbb{R}_+$. Let $\mathcal{F}_{\text{etrs}} = \{x : r \leq \|x\| \leq R,\ \|x - c\| \leq b^T x - a\}$ denote the feasible set of (4.6). Some examples of $\mathcal{F}_{\text{etrs}}$ with $x \in \mathbb{R}^2$ are shown in Figures 4.2–4.4. We are interested in the set $\mathcal{C}_{\text{etrs}} = \overline{\text{conv}}\{(x, X) : x \in \mathcal{F}_{\text{etrs}},\ X = xx^T\}$, since this would allow us to solve (4.6) by solving $\min\{H \bullet X + 2\, g^T x : (x, X) \in \mathcal{C}_{\text{etrs}}\}$ [21]. Using the techniques in Table (4.1), the

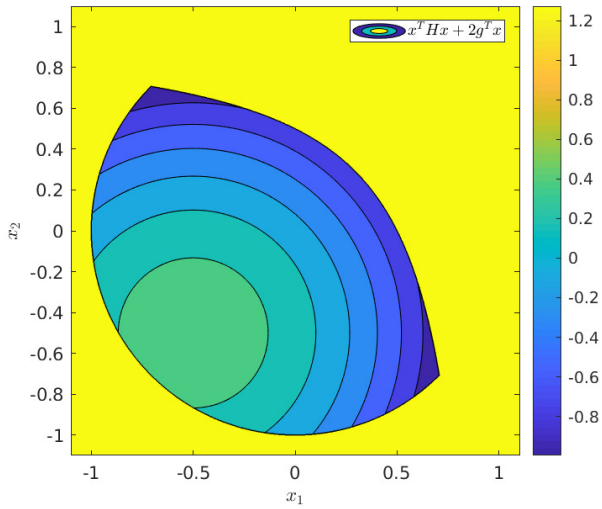


Figure 4.2: The feasible set $\mathcal{F}_{\text{etrs}}$ and objective of Example 1 in paper B. Yellow indicates infeasible points.

strongest relaxation of (4.6) is the Shor relaxation with the KSOC constraint enforced:

$$\begin{aligned}
\text{minimize} \quad & H \bullet X + 2\, g^T x && \text{(4.7a)}\\
\text{subject to} \quad & r^2 \leq \operatorname{tr}(X) \leq R^2 && \text{(4.7b)}\\
& \operatorname{tr}(X) - 2\, c^T x + c^T c \leq bb^T \bullet X - 2\, a\, b^T x + a^2 && \text{(4.7c)}\\
& 0 \leq b^T x - a && \text{(4.7d)}\\
& \begin{pmatrix} 1 & x^T \\ x & X \end{pmatrix} \succeq 0 && \text{(4.7e)}\\
& \begin{pmatrix} R & x^T \\ x & R\, I \end{pmatrix} \otimes \begin{pmatrix} b^T x - a & (x-c)^T \\ x - c & (b^T x - a) I \end{pmatrix} \succeq 0. && \text{(4.7f)}
\end{aligned}$$

We denote the feasible set of this relaxation $\mathcal{R}_{\text{shor}} \cap \mathcal{R}_{\text{ksoc}} \subseteq \mathbb{R}^n \times \mathcal{S}^n$.

In an effort to approximate $\mathcal{C}_{\text{etrs}}$ better, we derive a class of new valid inequalities for $\mathcal{F}_{\text{etrs}}$ which can be enforced as cuts in the lifted/relaxation space. The new inequalities are based on:

- Self-duality of the SOC and the fact that it is a convex cone ($\alpha, \beta \geq 0 \ \wedge \ x, y \in \text{SOC} \implies \alpha x + \beta y \in \text{SOC}$).

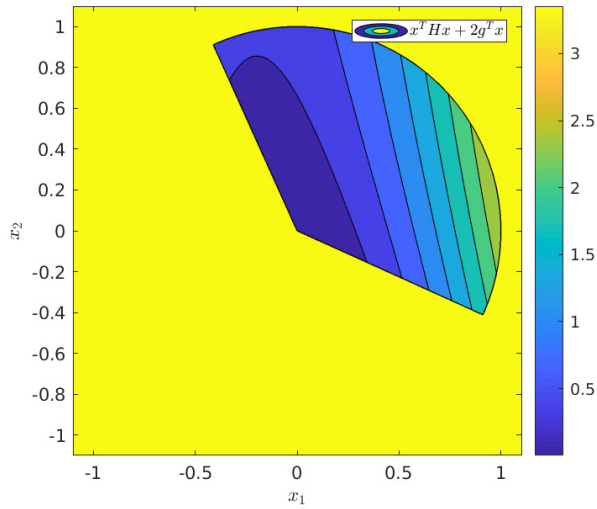


Figure 4.3: The feasible set $\mathcal{F}_{\text{etrs}}$ with $a = r = 0$, $c = 0$, $b = (2, 2)$, $R = 1$. The contours are for a randomly generated objective. Yellow indicates infeasible points.

- A pair of nonnegative functions $q(x)$ and $l(x)$, *i.e.*, the functions satisfy $q(x) \geq 0$ and $l(x) \geq 0$ for all $x \in \mathcal{F}_{\text{etrs}}$. We assume that q is a quadratic function and l is a linear function. These functions can be chosen, which is what makes the inequalities a class. In addition to the functions, we use a lower bound $[q + l]_{\min} \in \mathbb{R}_+$ on the sum of these functions, *i.e.*, $q(x) + l(x) \geq [q + l]_{\min}$ for all $x \in \mathcal{F}_{\text{etrs}}$.
- A constant $[c]_{\max} \in \mathbb{R}_+$ such that $rc^T x \|x\|^{-2} \leq [c]_{\max}$. Given a problem with data r, R, c, b, a, we can compute $[c]_{\max}$. Hence, this is a problem-dependent constant which can be computed as a preprocessing step before applying the cuts as we describe later.

We omit the details of the derivations (see paper B in Appendix B) but the derivation process is summarized in Figure 4.5. The resulting class of cuts can be stated as follows. Given $q(x) = x^T H_q x + g_q^T x + f_q$ and $l(x) = g_l^T x + f_l$, the following is a valid cut in the (x, X) space:

$$\begin{aligned}(r + R)R\left(H_q \bullet X + 2\, g_q^T x + f_q\right) + (r + R)\left(2\, g_l b^T \bullet X + (f_l b - 2a g_l)^T x - a f_l\right) \\ \geq [q + l]_{\min} \operatorname{tr}(X) + rR\left(H_q \bullet X + 2(g_q + g_l)^T x + (f_q + f_l)\right) \\ - \left(2\, g_l c^T \bullet X + f_l c^T x\right) - [c]_{\max} R(2\, g_l^T x + f_l). \end{aligned} \tag{4.8}$$

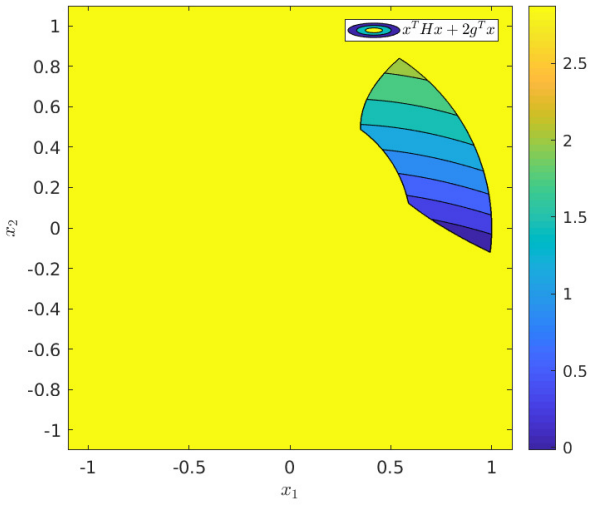


Figure 4.4: The feasible set $\mathcal{F}_{\text{etrs}}$ with $a = 0.2$,$r = 0.6$, $c = (0.5, 0.5)$, $b = (1, 0)$, $R = 1$. The contours are for a randomly generated objective. Yellow indicates infeasible points.

These cuts may look slightly complicated, but note that they are linear in (x, X). For fixed (x, X), the inequalities are also linear in H_q, g_q, f_q, g_l, f_l. This is important, since it allows us to *separate* the cuts dynamically, *i.e.*, given a point $(\bar{x}, \bar{X})$, we can check if this violates any of the cuts (for any choice of $q(x), l(x)$) in polynomial time. This means that we can "bootstrap" the cuts onto an SDP relaxation. Given a relaxation with objective function $\tilde{f}(x, X) = H \bullet X + 2\, g^T x$ and feasible set $\mathcal{R}$, define $\mathcal{R}_0 = \mathcal{R}$. Then the bootstrapping can be described by the following steps.

0. Let $k = 0$.
1. Solve $\min\{\tilde{f}(x, X) : (x, X) \in \mathcal{R}_k\}$ and denote the solution by $(\bar{x}_k, \bar{X}_k)$.
2. Solve the separation problem described in paper B to obtain a pair of functions q_{k+1}, l_{k+1}. (These are given by their coefficients H_q, g_q, f_q, g_l, f_l.) If (4.8) is not violated by any q, l, stop; otherwise, let K_{k+1} denote the set of (x, X) that satisfy (4.8) with q_{k+1} and l_{k+1}. Add cut: let $\mathcal{R}_{k+1} = \mathcal{R}_k \cap K_{k+1}$. Increment k by one, so $k = k + 1$. Go to step 1.

The process keeps going until we find a point $(\bar{x}_*, \bar{X}_*)$, which does not violate (4.8) for any valid q, l. This is the solution of the given relaxation with the class

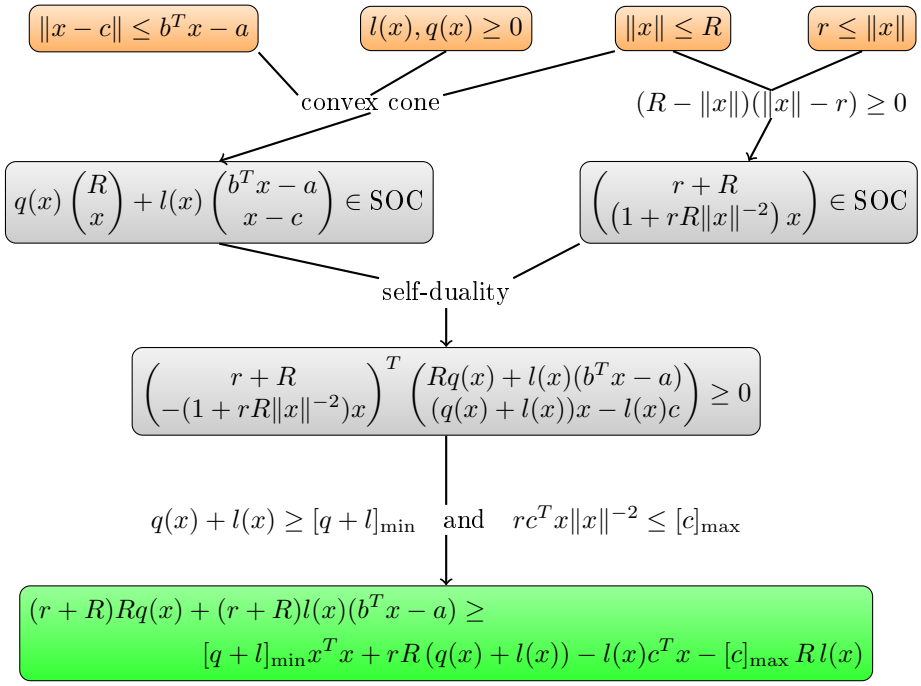


Figure 4.5: Outline of derivation of the new class of valid inequalities.

of new cuts added. The bootstrapping process is illustrated in Figure 4.6.

The computational cost of solving the separation problem depends on the relaxation that the cuts are bootstrapped to. Some implementation details can be found in Appendix E.

We can view the bootstrapping as iteratively tightening the relaxation by obtaining a sequence $K_1, K_2, \ldots, K_{n_{\text{cuts}}}$, where n_{cuts} denotes the number of cuts that are added. With this sequence we have that

$$\mathcal{R}_0 \supset \mathcal{R}_1 \supset \mathcal{R}_2 \supset \cdots \supset \mathcal{R}_{n_{\text{cuts}}} \supseteq \mathcal{C}_{\text{etrs}} \supseteq \mathcal{F}_{\text{lifted}}. \tag{4.9}$$

In addition to the application to OPF of these cuts, which is described in Section 5.5.2, we derived a class of valid SOC constraints for $\mathcal{F}_{\text{lifted}}$ when $\mathcal{F}$ is the intersection of the nonnegative orthant and the Euclidean ball. We conducted numerical experiments which demonstrate that the cuts strengthen both the Shor relaxation and the Shor relaxation with the KSOC constraint added. The cuts were particularly strong for the case when $c = 0$ and $a = r = 0$. In fact, we conjecture that the cuts added to the Shor-KSOC relaxation captures the

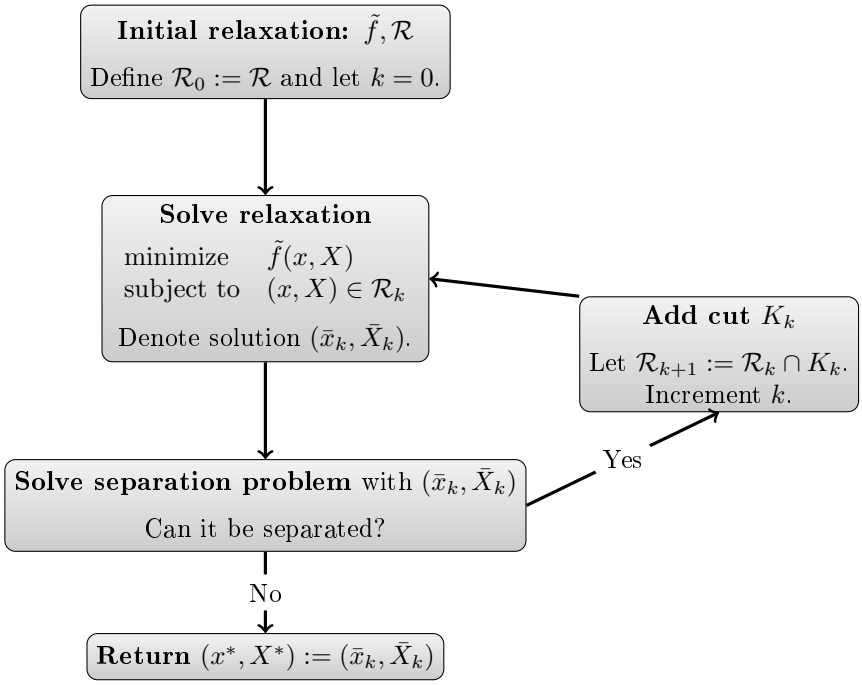


Figure 4.6: Bootstrapping procedure.

convex hull $\mathcal{C}_{\text{etrs}} = \overline{\text{conv}}\left\{(x, xx^T) : x \in \mathcal{F}_{\text{etrs}}\right\}$ when $\mathcal{F}_{\text{etrs}} = \{x \in \mathbb{R}^2 : \|x\| \leq R,\ \|x\| \leq b^T x\}$ for arbitrary $b \in \mathbb{R}^2$. This feasible set is depicted in Figure 4.3 for a given b.

4.2 Orthogonal Generalization

The inequalities developed in paper B use the self-duality of the SOC as part of the derivations. As we can see in Example 1 in the paper, the inequality derived from *just* the self-duality (which is included in the class of inequalities) can be effective. As opposed to some of the other derivations, the inequality derived from the self-duality is quadratic even when the Hessians of the SOCs involved are different. Hence, for a pair of general SOC constraints

$$\begin{pmatrix} b_1^T x - a_1 \\ H_1 x - c_1 \end{pmatrix}, \begin{pmatrix} b_2^T x - a_2 \\ H_2 x - c_2 \end{pmatrix} \in \text{SOC} \tag{4.10}$$

we can derive the valid quadratic inequality

$$\begin{pmatrix} b_1^T x - a_1 \\ -(H_1 x - c_1) \end{pmatrix}^T \begin{pmatrix} b_2^T x - a_2 \\ H_2 x - c_2 \end{pmatrix} \geq 0$$
$$\Updownarrow$$
$$x^T(b_1 b_2^T - H_1^T H_2)x + (H_1^T c_2 + c_1^T H_2 - a_2 b_1^T - a_1 b_2^T)x - c_1^T c_2 + a_1 a_2 \geq 0 \tag{4.11}$$

An observation about using the self-duality is that we have changed the sign "inside the norm" of one of the SOCs, *i.e.*, we use $\|-(H_1 x - c_1)\| \leq b_1^T x - a_1$ instead of $\|H_1 x - c_1\| \leq b_1^T x - a_1$. This is actually a specific choice for the following generalization.

Let $\mathcal{O}_n$ denote the space of $n \times n$ orthogonal matrices, and suppose $Q \in \mathcal{O}_n$. Then we have

$$\|x\| \leq R \iff \|Qx\| \leq R.$$

Using this we can generalize (4.11) to

$$\begin{pmatrix} b_1^T x - a_1 \\ Q^T(H_1 x - c_1) \end{pmatrix}^T \begin{pmatrix} b_2^T x - a_2 \\ H_2 x - c_2 \end{pmatrix} \geq 0$$
$$\Updownarrow$$
$$\begin{aligned} x^T(b_1 b_2^T + Q H_1^T H_2)x - (Q H_1^T c_2 + c_1^T Q H_2 - a_2 b_1^T - a_1 b_2^T)x \\ + c_1^T Q c_2 + a_1 a_2 \geq 0. \end{aligned} \tag{4.12}$$

Since we can choose Q freely, this is an infinite set of valid inequalities. We call (4.12) the *orthogonal generalization* of (4.11). Note that the choice $Q = -I$ recovers (4.11). This orthogonal generalization can also be applied to the "self-duality" step in Figure 4.5. This generalization is most interesting when Q can be chosen efficiently. Since the inequality (4.11) is linear in Q (for fixed (x, X)), we can compute the best Q at the cost of an SVD (see Appendix E.3).

We did some preliminary experimentation with the orthogonal generalization for a simpler version of the inequalities (4.8) and identified some cases where the orthogonal generalization did not improve the relaxation beyond the cuts (at that time). Some of this can be seen in Appendix E.3.

Chapter 5

Optimal Power Flow

The alternating current optimal power flow (ACOPF) problem is a fundamental problem in power systems engineering. In this chapter, we provide some background and discuss the modelling of the ACOPF problem, since it plays a role in all the papers—explicitly or as a motivation. We first describe the problem, the challenges, and the mathematical formulation that we have worked with. Then we describe how the papers can contribute to tackling the challenges.

Electricity is a commodity that is taken for granted in many parts of the world; when you plug your device into the wall you expect it to get charged. The power network that delivers this power is one of the largest human-made engineering system. The power network, also called the power grid, consists of a set of geographical locations and some power lines connecting these. A geographical location is called a node, or a bus, and is an injection/extraction point, where there is a demand for power and possibly a set of generators capable of delivering power. The power network consists of a transmission network where most of the transmission happens and a distribution network which connects the transmission network to the consumers. The distribution is usually radial, which means that there are no cycles in the network, so the topology of distribution networks is generally a tree. The goal of the OPF problem is to minimize the cost of power generation while meeting the demand in the network. This is achieved by determining a dispatch—how much power should each generator generate—and the voltages, which control how the power flows. The ACOPF problem

is NP hard [13, 56, 55]. Aside from the immense practical importance of the OPF problem, it is interesting from the perspective of convex relaxation due to the many modelling choices one can make; see [14] for a survey on the various formulations. This has led to many different convex relaxations [5, 15, 51, 47, 24, 67]. For a survey of these, we refer the reader to [68] or the two-part survey [61, 62]. A detailed description of power systems can be found in, *e.g.*, [38]. For an account of convex optimization problems in power systems, see, *e.g.*, [81].

5.1 Challenges

The OPF problem is hard to solve and the amount of money involved is huge [22]. Therefore, it is of interest to obtain a certificate of global optimality or a an upper bound on the suboptimality, which is exactly what a convex relaxation can provide. Empirical evidence shows that the SDP relaxation is very tight for many instances [55]. Here we define an instance as the combination of the network topology and a specific demand. However, for large network instances there has been concern about the tractability of solving the SDP relaxation within the required time frame, which is around 5–15 minutes [22]. This motivated the numerical study of the scalability of the SDP relaxation conducted in Paper A. We summarize and discuss this in Section 5.5.1.

The SDP relaxation for the OPF problem is exact for many instances, but for the instances where the relaxation is not exact, there is an interest in strengthening the relaxation. One approach for this is to use valid inequalities. The valid inequalities in Paper B have an application in OPF and we discuss this in Section 5.5.2. For other approaches to strengthening for OPF, see, *e.g.*, [40].

With the introduction of distributed energy resources (solar cells and other household generation) there has been an increasing interest in operating the distribution network in recent years [73]. The distribution network is usually radial (has no cycles) and for these instances the SDP relaxation has proven to be particularly effective [36]. Under different assumptions it has been proven that the relaxation will be exact [62, 55, 17, 67] but these assumptions are not always satisfied. From a practical point of view, it is probably not so important that the relaxation is exact for *all* radial networks. Instead, a system operator is probably more interested in a guarantee that a *particular* network instance has an exact relaxation. This can be loosely translated to guaranteeing exactness for a problem with specific structure and specific data (without solving the relaxation). This is the topic of paper C, where we consider homogeneous QCQPs with forest structure; we can bring the OPF problem into the form of a homogeneous QCQP, and for distribution networks we have the desired forest

structure. We discuss the QCQP formulation of OPF in Section 5.3 and describe how paper C relates to OPF in Section 5.5.3.

5.2 Mathematical Model

Mathematically, we model the power network as a graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$, where the vertices are the buses and the edges are the power lines. At each bus we have a complex variable describing the voltage and a complex variable describing the current. We denote the vector of voltages by $v \in \mathbb{C}^n$ and the vector of currents by $i \in \mathbb{C}^n$. The formulation of the OPF that we consider is

$$
\begin{aligned}
&\text{minimize} && \sum_{g \in G} f_g(p_g) && && \text{(5.1a)}\\
&\text{subject to} && i_k^* v_k = \sum_{g \in G_k} s_g - S_k^d && k = 1, \ldots, n && \text{(5.1b)}\\
& && \underline{V}_k \leq |v_k| \leq \overline{V}_k && k = 1, \ldots, n && \text{(5.1c)}\\
& && \underline{S}_g \leq s_g \leq \overline{S}_g && \forall g \in G && \text{(5.1d)}\\
& && |F_{kl}(v)| \leq \overline{F}_{kl} && (k,l) \in \mathcal{L}^{\text{fl}} && \text{(5.1e)}\\
& && \underline{\phi}_{kl} \leq \angle(v_k v_l^*) \leq \overline{\phi}_{kl} && (k,l) \in \mathcal{L}^{\text{pa}} && \text{(5.1f)}\\
& && i = Yv && && \text{(5.1g)}
\end{aligned}
$$

where the variables are $i, v \in \mathbb{C}^n$ and $s_g \in \mathbb{C}$ $(g \in G)$. The data is described in Table 5.1. The function $F_{kl}(v)$ describes the flow from bus k to bus l. Note that for a power line (k,l) with a flow (thermal) limit, we have that $(k,l), (l,k) \in \mathcal{L}^{\text{fl}}$ and that $\overline{F}_{kl} = \overline{F}_{lk}$, so (5.1e) covers flow in both directions. There are several types of flow that can be used [92]; current flow (linear in v), apparent power flow (quadratic in v), or real power flow (quadratic in v). In paper A, we consider the apparent power flow, which results in an SOC constraint in the relaxation. To obtain a QCQP formulation in Section 5.3, we use the real power flow as in [55, 17]. The network topology (the connections between the vertices) is especially captured by the admittance matrix Y, since its sparsity graph is the network graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$.

It is common practice to eliminate the currents, i, from the problem using Ohm's law (5.1g):

$$
i_k^* = v^H Y^H e_k.
$$

Table 5.1: Sets and data describing a power network.

Notation	Description
$Y \in \mathcal{H}^n$	the admittance matrix of the network
$S_k^d \in \mathbb{C}$	complex power demand at bus k
$S_k^{max} \in \mathbb{C}$	maximum power production at bus k
$S_k^{min} \in \mathbb{C}$	minimum power production at bus k
$U_k \in \mathbb{R}_+$	maximum voltage magnitude at bus k
$L_k \in \mathbb{R}_+$	minimum voltage magnitude at bus k
$F_{kl} \in \mathbb{R}_+$	upper bound on the flow from bus k to bus l
$\overline{\phi}_{kl} \in (-\pi, \pi)$	maximum voltage angle difference between bus k and l
$\underline{\phi}_{kl} \in (-\pi, \pi)$	minimum voltage angle difference between bus k and l
$n = \lvert\mathcal{V}\rvert$	Number of buses
$\mathcal{L}^{\text{fl}} \subseteq \mathcal{E}$	Set of power lines with a limit on the power flow
$\mathcal{L}^{\text{pa}} \subseteq \mathcal{E}$	Set of power lines with a limit on phase angle
G	Set of generators
$G_k \subseteq G$	Set of generators at vertex k

Using $v_k = e_k^T v$, the problem becomes

$$\begin{aligned}
\text{minimize} \quad & \sum_{g \in G} f_g(p_g) && && \text{(5.2a)}\\
\text{subject to} \quad & v^H Y^H e_k e_k^T v = \sum_{g \in G_k} s_g - S_k^d && k = 1, \ldots, n && \text{(5.2b)}\\
& \underline{V}_k \leq |v_k| \leq \overline{V}_k && k = 1, \ldots, n && \text{(5.2c)}\\
& \underline{S}_g \leq s_g \leq \overline{S}_g && \forall g \in G && \text{(5.2d)}\\
& |F_{kl}(v)| \leq \overline{F}_{kl} && (k,l) \in \mathcal{L}^{\text{fl}} && \text{(5.2e)}\\
& \underline{\phi}_{kl} \leq \angle(v_k v_l^*) \leq \overline{\phi}_{kl} && (k,l) \in \mathcal{L}^{\text{pa}} && \text{(5.2f)}
\end{aligned}$$

The model (5.1) is important for a lifting that we describe in Section 5.4 but the model (5.2) is more convenient for an SDP relaxation.

The constraints (5.2b) and (5.2d) are complex constraints. Denote the real power of generator g by p_g and the reactive power by q_g, so that $s_g = p_g + \imath q_g$. For each bus $k = 1, \ldots, n$, let P_k be the real power demand, let Q_k be the reactive power demand, and define the two matrices $\overline{Y}_k = \frac{1}{2}(Y^H e_k e_k^T + e_k e_k^T Y)$ and $\widetilde{Y}_k = \frac{1}{2\imath}(Y^H e_k e_k^T + e_k e_k^T Y)$. Then the power balance (5.1b) at bus k can be split in the real (active) and imaginary (reactive) power

$$v^H \overline{Y}_k v = \sum_{g \in G_k} p_g - P_k^d, \quad v^H \widetilde{Y}_k v = \sum_{g \in G_k} q_g - Q_k^d, \tag{5.3}$$

and the generator limits can be expressed as

$$\underline{P}_g \leq p_g \leq \overline{P}_g, \quad \underline{Q}_g \leq q_g \leq \overline{Q}_g, \tag{5.4}$$

where $\overline{S}_g = \overline{P}_g + \imath\overline{Q}_g$ and $\underline{S}_g = \underline{P}_g + \imath\underline{Q}_g$.

The phase angle difference constraint can be reformulated as follows. A complex number v_k can be expressed in terms of its magnitude and angle as

$$v_k = |v_k| e^{\imath(\angle v_k)} = |v_k| \left(\cos(\angle v_k) + \imath \sin(\angle v_k)\right), \tag{5.5}$$

where e is Euler's number. Hence, tangent of the angle difference between v_k and v_l can be calculated from the complex numbers as

$$\tan(\angle v_k - \angle v_l) = \frac{\operatorname{Im}(v_k v_l^*)}{\operatorname{Re}(v_k v_l^*)}. \tag{5.6}$$

When $-\pi/2 < \underline{\phi}_{kl} \leq \overline{\phi}_{kl} < \pi/2$, we can express the phase angle constraint (5.1f) as

$$\begin{aligned} \tan(\phi_{kl}) \leq & \frac{\operatorname{Im}(v_k v_l^*)}{\operatorname{Re}(v_k v_l^*)} \leq \tan(\phi_{kl}) \\ \iff \tan(\phi_{kl}) \operatorname{Re}(v_k v_l^*) \leq & \operatorname{Im}(v_k v_l^*) \leq \tan(\phi_{kl}) \operatorname{Re}(v_k v_l^*). \end{aligned} \tag{5.7}$$

With the described transformations, we can express the OPF as

$$\begin{aligned}
\text{minimize} \quad & \sum_{g \in G} f_g(p_g) && && \text{(5.8a)}\\
\text{subject to} \quad & \overline{Y}_k \bullet W = \sum_{g \in G_k} p_g - P_k^d && k = 1, \ldots, n && \text{(5.8b)}\\
& \widetilde{Y}_k \bullet W = \sum_{g \in G_k} q_g - q_k^d && k = 1, \ldots, n && \text{(5.8c)}\\
& \underline{V}_k^2 \leq W_{kk} \leq \overline{V}_k^2 && k = 1, \ldots, n && \text{(5.8d)}\\
& \underline{P}_g \leq p_g \leq \overline{P}_g && \forall g \in G && \text{(5.8e)}\\
& \underline{Q}_g \leq q_g \leq \overline{Q}_g && \forall g \in G && \text{(5.8f)}\\
& |F_{kl}(W_{kl})| \leq \overline{F}_{kl} && (k,l) \in \mathcal{L}^{\text{fl}} && \text{(5.8g)}\\
& \tan(\phi_{kl}) \operatorname{Re}(W_{kl}) \leq \operatorname{Im}(W_{kl}) \leq \tan(\phi_{kl}) \operatorname{Re}(W_{kl}) && (k,l) \in \mathcal{L}^{\text{pa}} && \text{(5.8h)}\\
& W = vv^H && && \text{(5.8i)}
\end{aligned}$$

where the variables are $v \in \mathbb{C}^n$ and $W = W^H \in \mathbb{C}^{n \times n}$. Assume that the objective function $f_g(p_g)$ is convex, then a cone LP relaxation can be obtained by replacing $W = vv^H$ with $W \succeq 0$. Note that (5.8g) is a second order cone constraint when the flow constraint is on the apparent power. In the next section we will describe some assumptions under which the OPF problem can be modelled as a QCQP.

5.3 OPF as a Homogeneous QCQP

In the following we describe how the OPF can be formulated as a homogeneous QCQP. We will discuss the objective and constraints of (5.2) one at a time in order of appearance. We will assume that each bus has at most one generator, since this allows us to eliminate the generation variables by using the power balance constraints (5.1b), and that the objective is either linear in the active power or it is the power loss in the network which is quadratic in the voltages.

Objective function: Let c_k be the (linear) cost of active generation at node k, then we can consider the objective

$$\sum_{k=1}^{n} c_k(p_k - P_k^d) = \sum_{k=1}^{n} c_k v^H \overline{Y}_k v - \sum_{k=1}^{n} c_k P_k^d = v^H C_{\text{gen}} v - \gamma. \tag{5.9}$$

where $C_{\text{gen}} = \sum_{k=1}^{n} c_k \overline{Y}_k$ and $\gamma = \sum_{k=1}^{n} c_k P_k^d$.

Another objective, which is more obviously quadratic in the voltages is power loss, where the objective is

$$v^H C_{\text{loss}} v, \tag{5.10}$$

where $C_{\text{loss}} = (Y + Y^H)/2$.

Power balance and generation limits: When there is at most one generator at bus k, the power balance (5.1b) can be expressed as

$$v^H Y^H e_k e_k^T v = s_k - S_k^d. \tag{5.11}$$

Define the new bounds

$$P_k^{\min} = \underline{P}_k - P_k^d, \; P_k^{\max} = \overline{P}_k - P_k^d, \tag{5.12}$$

$$Q_k^{\min} = \underline{Q}_k - Q_k^d, \; Q_k^{\max} = \overline{Q}_k - Q_k^d. \tag{5.13}$$

Using (5.3), we can substitute the power balance into the generation limits:

$$P_k^{\min} \le v^H \overline{Y}_k v \le P_k^{\max}, \quad Q_k^{\min} \le v^H \widetilde{Y}_k v \le Q_k^{\max}. \tag{5.14}$$

Note that for buses with no generation capacity, the upper and lower bounds are the same ($P_k^{\min} = P_k^{\max}$ and $Q_k^{\min} = Q_k^{\max}$) and will essentially result in an equality constraint in the final QCQP.

Voltage magnitude bounds The nodal voltages are constrained by

$$L_k \le |v_k| \le U_k \iff L_k^2 \le v^H \overline{E}_{kk} v \le U_k^2, \tag{5.15}$$

where $\overline{E}_{kk} = e_k e_k^T$.

Phase angle difference: We have already seen that under the assumption that $-\pi/2 < \underline{\phi}_{kl} \le \overline{\phi}_{kl} < \pi/2$, we can write the phase angle difference constraints (5.1f) as

$$\tan(\underline{\phi}_{kl}) \operatorname{Re}(v_k v_l^*) \le \operatorname{Im}(v_k v_l^*) \le \tan(\overline{\phi}_{kl}) \operatorname{Re}(v_k v_l^*) \tag{5.16}$$

We can write the right inequality as

$$\operatorname{Im}(v^H e_l e_k^T v) - \tan(\overline{\phi}_{kl}) \operatorname{Re}(v^H e_l e_k^T v) \leq 0. \tag{5.17}$$

Define $\overline{E}_{kl} = \frac{1}{2}(e_l e_k^T + e_k e_l^T)$ and $\widetilde{E}_{kl} = \frac{1}{2i}(e_l e_k^T - e_k e_l^T)$. Then we can write this as

$$v^H(\widetilde{E}_{kl} - \tan(\overline{\phi}_{kl})\overline{E}_{kl})v \leq 0. \tag{5.18}$$

Similarly, from the lower bound we get

$$v^H(\tan(\underline{\phi}_{kl})\overline{E}_{kl} - \widetilde{E}_{kl})v \leq 0. \tag{5.19}$$

Define

$$\overline{\Phi}_{kl} = \widetilde{E}_{kl} - \tan(\overline{\phi}_{kl})\overline{E}_{kl},\ \underline{\Phi}_{kl} = \tan(\underline{\phi}_{kl})\overline{E}_{kl} - \widetilde{E}_{kl}, \tag{5.20}$$

so the constraints become

$$v^H \overline{\Phi}_{kl} v \leq 0,\ v^H \underline{\Phi}_{kl} v \leq 0. \tag{5.21}$$

Line flows: Using the active power flow in (5.1e), we may express $F_{kl}(v) = v^H T_{kl} v$, where T_{kl} is a given matrix [92]. Then, the line flow constraints becomes

$$|v^H T_{kl} v| \leq \overline{F}_{kl} \iff -\overline{F}_{kl} \leq v^H T_{kl} v \leq \overline{F}_{kl}. \tag{5.22}$$

QCQP formulation: Let $C \in \{C_{loss}, C_{gen}\}$, then we can write the OPF problem (5.1) as the QCQP

$$\begin{aligned}
\text{minimize} \quad & v^H C v && && \text{(5.23a)}\\
\text{subject to} \quad & P_k^{\min} \leq v^H \overline{Y}_k v \leq P_k^{\max}, && k = 1, \ldots, n && \text{(5.23b)}\\
& Q_k^{\min} \leq v^H \widetilde{Y}_k v \leq Q_k^{\max}, && k = 1, \ldots, n && \text{(5.23c)}\\
& L_k \leq v^H J_k v \leq U_k, && k = 1, \ldots, n && \text{(5.23d)}\\
& v^H \overline{\Phi}_{kl} v \leq 0, && (k,l) \in \mathcal{L}^{\text{pa}} && \text{(5.23e)}\\
& v^H \underline{\Phi}_{kl} v \leq 0, && (k,l) \in \mathcal{L}^{\text{pa}} && \text{(5.23f)}\\
& -\overline{F}_{kl} \leq v^H T_{kl} v \leq \overline{F}_{kl}, && (k,l) \in \mathcal{L}^{\text{fl}}. && \text{(5.23g)}
\end{aligned}$$

From this, we can obtain an SDP relaxation with the lifting procedure described in Section 2.1. This is a homogeneous QCQP, which will have tree structure when the admittance matrix Y has tree structure.

5.4 Current-Voltage Relaxation

The main contributor to the large solution time of the semidefinite relaxation, especially for larger networks, is the cone constraint $W \succeq 0$. Therefore, many relaxations proposed in the literature lean more towards the SOCP relaxation than the SDP relaxation; the SOCP relaxation can be obtained from a branch flow model of the OPF [61] or by relaxing the constraint $W \succeq 0$ to positive semidefiniteness of its 2×2 minors. In the spirit of avoiding the constraint $W \succeq 0$ one could be tempted to do the following: Instead of eliminating the currents (going from (5.1) to (5.2)), we can keep the currents and relax (5.1) by the lifting technique described in Section 2.1. We describe this in the following. For the sake of presentation we do not consider the flow and phase angle difference constraints.

Consider the OPF problem (5.1). For each bus we define the vector

$$z_k = \begin{bmatrix} v_k \\ i_k \end{bmatrix}, \quad k = 1, \dots, n, \tag{5.24}$$

and the matrix variable

$$Z_k = z_k z_k^H = \begin{bmatrix} v_k v_k^* & v_k i_k^* \\ i_k v_k^* & i_k i_k^* \end{bmatrix} = \begin{bmatrix} |v_k|^2 & v_k i_k^* \\ i_k v_k^* & |i_k|^2 \end{bmatrix} \tag{5.25}$$

Let $E_{ij} = e_i e_j^T$ be a 2×2 matrix with a one in row i column j and zeros otherwise. Then we can write the OPF problem (5.1) as

$$\begin{aligned}
\text{minimize} \quad & \sum_{g \in G} f_g(p_g) && && \text{(5.26a)} \\
\text{subject to} \quad & E_{12} \bullet Z_k = \sum_{g \in G_k} s_g - S_k^d && k = 1, \dots, n && \text{(5.26b)} \\
& \underline{V}_k^2 \le E_{11} \bullet Z_k \le \overline{V}_k^2 && k = 1, \dots, n && \text{(5.26c)} \\
& \underline{S}_g \le s_g \le \overline{S}_g && \forall g \in G && \text{(5.26d)} \\
& i_k = \sum_{l=1}^{n} Y_{kl} v_l, \quad k = 1, \dots, n && && \text{(5.26e)} \\
& Z_k = \begin{bmatrix} v_k \\ i_k \end{bmatrix} \begin{bmatrix} v_k \\ i_k \end{bmatrix}^H, \quad k = 1, \dots, n && && \text{(5.26f)}
\end{aligned}$$

A relaxation of this is readily obtained by relaxing the last constraints to

$$Z_k \succeq \begin{bmatrix} v_k \\ i_k \end{bmatrix} \begin{bmatrix} v_k \\ i_k \end{bmatrix}^H \iff \begin{bmatrix} 1 & v_k^* & i_k^* \\ v_k & [Z_k]_{11} & [Z_k]_{12} \\ i_k & [Z_k]_{21} & [Z_k]_{22} \end{bmatrix} \succeq 0, \quad k = 1, \dots, n. \tag{5.27}$$

If we look at the constraints of problem (5.26), it is evident that the currents and voltages only appear in the constraints (5.26e) and (5.26f). If $Z_k \succeq 0$, then $i_k = v_k = 0$ is feasible. Hence, the variables have been decoupled from the problem and can essentially be eliminated.

If we eliminate v_k and i_k from the problem, it reduces to

$$\begin{aligned}
\text{minimize} \quad & \sum_{g \in G} f_g(p_g) && && \text{(5.28a)}\\
\text{subject to} \quad & E_{12} \bullet Z_k = \sum_{g \in G_k} s_g - S_k^d && k = 1, \dots, n && \text{(5.28b)}\\
& \underline{V}_k^2 \leq E_{11} \bullet Z_k \leq \overline{V}_k^2 && k = 1, \dots, n && \text{(5.28c)}\\
& \underline{S}_g \leq s_g \leq \overline{S}_g && \forall g \in G && \text{(5.28d)}\\
& Z_k \succeq 0, \quad k = 1, \dots, n. && && \text{(5.28e)}
\end{aligned}$$

Note that the admittance matrix is also eliminated and that $[Z_k]_{22}$ only appears in the constraint $Z_k \succeq 0$. For the problem to be bounded we need an upper bound on $[Z_k]_{22}$, which corresponds to the squared current magnitude at bus k. One way to obtain this is described in Appendix F.1.

This approach is very similar to a diagonalization approach, where the rank-1 matrices $\overline{Y}_k$ and $\check{Y}_k$ are used to define new variables. We outline this in Appendix F.2. The diagonalization approach inspired the current-voltage relaxation and suffers from the same drawback that the constraints become decoupled. These relaxations are not very useful in approximating the original problem, but they illustrate the point that diagonalization before lifting is not a good approach for homogeneous QCQPs. This is also outlined in the discussion in paper C.

5.5 Contributions

In this section, we describe the contributions of Papers A–C in relation to the OPF problem. Recall the challenges described in the beginning of this chapter.

5.5.1 Paper A

In paper A, we investigate the scalability and robustness of the SDP relaxation of the OPF problem. To this end, we formulate the OPF problem as a cone LP

(similar to (5.8)) of the form

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & Ax = b \\ & x \in \mathcal{K}, \end{array} \tag{5.29}$$

where the cone $\mathcal{K}$ is a cartesian product of the nonnegative orthant, a number of second order cones, and the cone of Hermitian positive semidefinite matrices:

$$\mathcal{K} = \mathbb{R}^n_+ \times \underbrace{\mathcal{K}^3_q \times \cdots \times \mathcal{K}^3_q}_{n_q} \times \mathcal{K}^n_h.$$

The aim of formulating the OPF in this standard form is to facilitate a comparison of the robustness and scalability of different solvers. The computational bottleneck of solving (5.29) for large n is usually the Hermitian positive semidefinite cone $\mathcal{K}^n_h$, but due to the sparsity of the OPF problem we can apply the conversion of [35], which essentially decomposes the large cone into smaller cones at the cost of some additional equality constraints. The converted problem can be formulated as

$$\begin{array}{ll} \text{minimize} & \tilde{c}^T x \\ \text{subject to} & \tilde{A}x = b \\ & \tilde{E}x = 0 \\ & x \in \tilde{\mathcal{K}}, \end{array} \tag{5.30}$$

where

$$\tilde{\mathcal{K}} = \mathbb{R}^n_+ \times \underbrace{\mathcal{K}^3_q \times \cdots \times \mathcal{K}^3_q}_{n_q} \times \mathcal{K}^{r_1}_h \times \cdots \times \mathcal{K}^{r_m}_h.$$

This is intended to mitigate the computational cost of the large cone $\mathcal{K}^n_h$. There are different ways to make the conversion and there is a trade-off between the number of equalities that are introduced and the size of the blocks.

In our experiments, we solved the semidefinite relaxation with five different solvers. Comparing the solvers, we found that MOSEK was generally fastest and most robust; it solved all test instances to the given accuracy. In Figure 5.1 we see the solver time of MOSEK for the OPF problem for different networks. We can see that the solver time is superlinear in the number of buses, so the increase in solver time is not as large as one could expect for an SDP. A more theoretical account of this behavior can be found in [90]. If we instead plot the time against the largest clique in the network, seen in Figure 5.2, we see that this is more likely the computational bottleneck of solving the semidefinite relaxation of the OPF.

The main takeaways of the study are that the SDP relaxation can be a tractable relaxation for power networks of a fairly large size; the relaxation was solved in

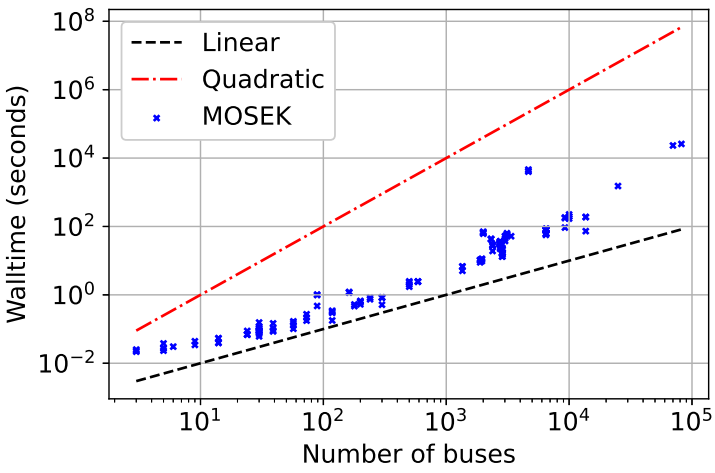


Figure 5.1: MOSEK solver times compared to the number of buses in the network (log-log). Tendency lines for linear and quadratic dependency plotted.

less than 10 minutes for all but one case with less than 25,000 buses. Moreover, the results demonstrate that solution times can be significantly improved by manually constructing the relaxation compared to the use of modelling tools.

5.5.2 Paper B

An interesting observation about the OPF problem is that it does not involve any linear terms in the voltages in many formulations, so in essence we only care about the quadratics. Since the voltages are complex this corresponds to only caring about the magnitude and the angle between different voltages. Denote the feasible set of (5.8) by

$$\mathcal{F}_{\text{OPF}} = \{W : W \text{ satisfies (5.8b)–(5.8i)}\}. \tag{5.31}$$

Then we would like to derive valid inequalities for this feasible set. One way to do this is by considering a pair of connected buses with their voltage magnitude and phase angle difference constraints. This is done by Chen *et al.* [23] where

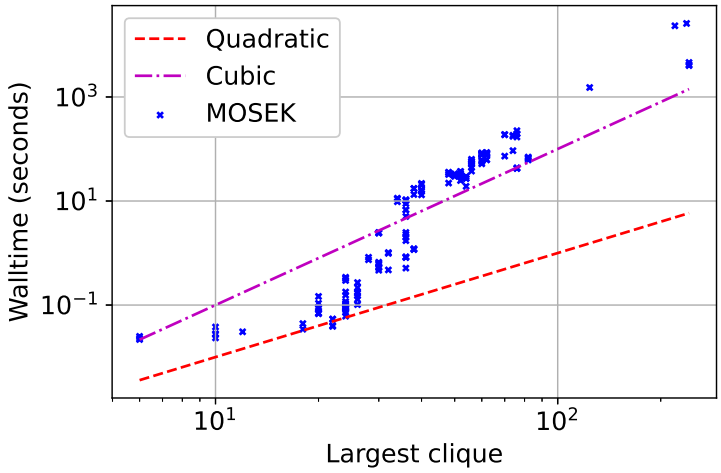


Figure 5.2: MOSEK solver times compared to the largest clique $s_{\max}$ after conversion (log-log). Tendency lines for quadratic and cubic dependency plotted.

they consider the set

$$L_{jj} \leq W_{jj} \leq U_{jj} \quad \forall\, j = 1, 2 \tag{5.32a}$$
$$L_{12}W_{12} \leq T_{12} \leq U_{12}W_{12} \tag{5.32b}$$
$$W_{12} \geq 0 \tag{5.32c}$$
$$W_{11}W_{22} = W_{12}^2 + T_{12}^2 \tag{5.32d}$$

where the four variables are $(W_{11}, W_{22}, W_{12}, T_{12}) \in \mathbb{R}^4$ and the data $L = (L_{11}, L_{22}, L_{12})$ and $U = (U_{11}, U_{22}, U_{12})$ satisfy $L \leq U$ and $L_{jj} \geq 0$ for $j = 1, 2$. Chen *et al.* prove that the convex hull of this set is captured by the natural SDP relaxation (relaxing the last constraint to $W_{11}W_{22} \geq W_{12}^2 + T_{12}^2$) intersected with a pair of linear inequalities. In paper B, we show that these linear inequalities can be seen as a special case of the larger class of valid inequalities that we derive. In particular, we show that they are equivalent to a pair of valid inequalities for the set

$$\mathcal{F}_{\mathrm{W}} = \left\{ x \in \mathbb{R}^3 : \begin{array}{c} \sqrt{L_{11}} \leq \left\| \binom{x_1}{x_2} \right\| \leq \sqrt{U_{11}} \\ \left\| \binom{x_1}{x_2} \right\| \leq b_1 x_1 + b_2 x_2 \\ \sqrt{L_{22}} \leq x_3 \leq \sqrt{U_{22}} \end{array} \right\} \tag{5.33}$$

The details can be seen in the paper. Loosely speaking, the set $\mathcal{F}_{\mathrm{W}}$ can be viewed as considering the following set

$$\begin{aligned}\underline{V}_1 \leq |v_1| \leq \overline{V}_1\\ \underline{V}_2 \leq |v_2| \leq \overline{V}_2\\ \underline{\phi}_{12} \leq \angle(v_1 v_2^*) \leq \overline{\phi}_{12},\end{aligned}$$

which is a subset of $\mathbb{C}^2$ and fixing the angle of one of the voltages which takes us to $\mathbb{R}^3$. An illustration of the set $\mathcal{F}_{\mathrm{W}}$ can be seen in Figure 5.3.

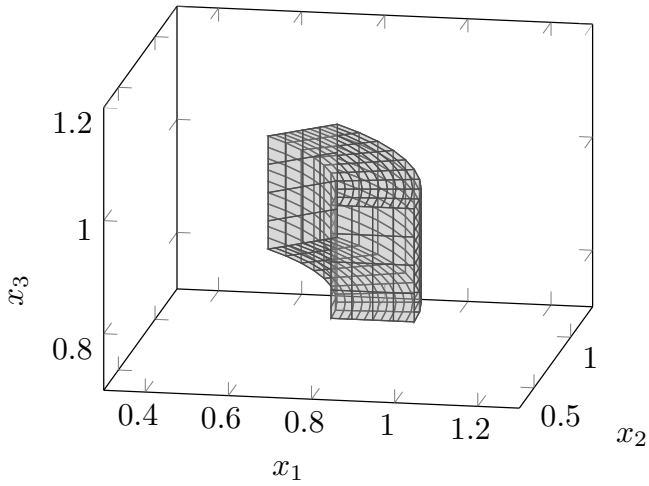


Figure 5.3: The set $\mathcal{F}_{\mathrm{W}}$ (5.33).

5.5.3 Paper C

As described in Section 5.3, the OPF problem can be formulated as a complex-valued QCQP. Paper C is motivated by the distribution networks, which are radial. In paper C, we consider a specific structure but the conditions also depend on the data in the problem, so that exactness can be checked for a specific distribution network.

In the paper we consider homogeneous QCQPs, but as we can see in problem (5.23), the QCQP formulation of the OPF has some additional structure: most of the quadratic terms have a lower and an upper bound. We call this type of

constraint a quadratic interval constraint, since the quadratic has to be within a specified interval. Therefore, it is sensible to consider the feasibility systems for this particular structure:

$$\begin{aligned} &\underset{x\in\mathcal{D}^n}{\text{minimize}} \;\; x^H A_0 x \\ &\text{subject to} \;\; \underline{b}_k \leq x^H A_k x \leq \overline{b}_k, \quad k = 1, \ldots, m. \end{aligned} \tag{5.34}$$

For this problem the dual problem (3.3) takes the form

$$\begin{aligned} &\text{minimize} \;\; \sum_{k=1}^{m} (\lambda_k \overline{b}_k - \mu_k \underline{b}_k) \\ &\text{subject to} \;\; A_0 + \sum_{k=1}^{m} (\lambda_k - \mu_k) A_k \succeq 0 \\ &\qquad\qquad \lambda, \mu \succeq 0. \end{aligned} \tag{5.35}$$

For the feasibility systems we disregard the objective, so if we define

$$\Omega_{\text{quad}-\text{int}} = \left\{ (\nu, Y) : \begin{array}{l} Y = A_0 + \sum_{k=1}^{m} \nu_k A_k \\ Y \succeq 0 \end{array} \right\},$$

we can formulate the feasibility systems (3.7) for (5.34) as

$$\Omega_{\text{quad}-\text{int}} \cap \mathbf{O}_{ij},$$

and the exactness condition (3.8) accordingly. Compared to the usual feasibility systems we do not have the constraint $\nu \succeq 0$.

Chapter 6

Conclusion

In this thesis, we have provided some background for convex relaxation and described the research that has been done throughout the project. The main contributions of the project can be summarized as:

Paper A We have conducted a numerical study of the semidefinite relaxation of the optimal power flow problem. The study compares different solvers and shows experimentally that the formulation of the problem (the modelling) has a large impact on the time it takes to solve the relaxation and on the robustness of the solvers. The study demonstrates that the semidefinite relaxation can be solved within minutes for networks with up to 10.000 buses. Hence, the semidefinite relaxation could be used as a complement to existing methods for solving the OPF problem, even in larger networks.

Paper B We have derived a new class of valid inequalities for an extended trust region subproblem. These inequalities can be used to iteratively tighten a semidefinite relaxation and improve relaxations for this problem. The inequalities are derived for a specific structure of the feasible set and can be applied whenever this structure is present. The derivation of these inequalities adds to the existing techniques for obtaining stronger relaxations.

Paper C We have proposed an exactness condition for the class of homogeneous quadratically constrained quadratic programs with forest structure.

This condition takes the numerical data of the problem into account, and can be used to guarantee exactness for a subclass of problems (with fixed data matrices but any data vector as described in Section 3.1 and in paper C). We propose a framework for guaranteeing exactness when some of the data in the instance is uncertain.

Some opportunities for future research based on this project include:

- Proving the conjectures of paper B, that the relevant convex hull is captured by the Shor relaxation intersected with the KSOC cut and the new cuts for a specific feasible set.
- Investigating the orthogonal inequalities in Section 4.2.
- Applying the framework of paper C to distribution networks of the OPF problem as discussed in Section 5.5.3 and perhaps to other applications.
- It would be interesting to combine the exactness analysis of paper C with the strengthening techniques of Paper B or other techniques. For larger problems it may not be tractable to use all valid inequalities, so it may be necessary to choose which constraints to use. Perhaps the exactness analysis could be used to determine this.
- In a similar vein, the exactness results of Paper C are for the standard semidefinite relaxation. It would be interesting to obtain similar results for strengthened relaxations.

Bibliography

[1] Emmanuel Abbe, Afonso S. Bandeira, and Georgina Hall. "Exact Recovery in the Stochastic Block Model". *IEEE Transactions on Information Theory* 62.1 (Jan. 2016), pp. 471–487. DOI: 10.1109/tit.2015.2490670.

[2] Farid Alizadeh. "Interior Point Methods in Semidefinite Programming with Applications to Combinatorial Optimization". *SIAM Journal on Optimization* 5.1 (Feb. 1995), pp. 13–51. DOI: 10.1137/0805002.

[3] Kurt M. Anstreicher. "Semidefinite programming versus the reformulation-linearization technique for nonconvex quadratically constrained quadratic programming". *Journal of Global Optimization* 43.2-3 (Nov. 2008), pp. 471–484. DOI: 10.1007/s10898-008-9372-0.

[4] Kurt M. Anstreicher. "Kronecker product constraints with an application to the two-trust-region subproblem". *SIAM Journal on Optimization* 27.1 (2017), pp. 368–378. DOI: 10.1137/16M1078859.

[5] Xiaoqing Bai et al. "Semidefinite programming for optimal power flow problems". *International Journal of Electrical Power & Energy Systems* 30.6-7 (July 2008), pp. 383–392. DOI: 10.1016/j.ijepes.2007.12.003.

[6] Egon Balas, Sebastián Ceria, and Gérard Cornuéjols. "A lift-and-project cutting plane algorithm for mixed 0–1 programs". *Mathematical Programming* 58.1-3 (Jan. 1993), pp. 295–324. DOI: 10.1007/bf01581273.

[7] Afonso S. Bandeira. "Convex Relaxations for Certain Inverse Problems on Graphs". PhD thesis. Princeton University, 2015.

[8] Xiaowei Bao, Nikolaos V. Sahinidis, and Mohit Tawarmalani. "Semidefinite relaxations for quadratically constrained quadratic programming: A review and comparisons". *Mathematical Programming* 129.1 (May 2011), pp. 129–157. DOI: 10.1007/s10107-011-0462-2.

[9] Amir Beck and Yonina C. Eldar. "Strong Duality in Nonconvex Quadratic Optimization with Two Quadratic Constraints". *SIAM Journal on Optimization* 17.3 (Jan. 2006), pp. 844–860. DOI: 10.1137/050644471.

[10] Pietro Belotti et al. "Mixed-integer nonlinear optimization". *Acta Numerica* 22 (Apr. 2013), pp. 1–131. DOI: 10.1017/s0962492913000032.

[11] Aharon Ben-Tal and Arkadi Nemirovski. *Lectures on Modern Convex Optimization*. Society for Industrial and Applied Mathematics, Jan. 2001. DOI: 10.1137/1.9780898718829.

[12] Daniel Bienstock and Alexander Michalka. "Polynomial Solvability of Variants of the Trust-Region Subproblem". In: *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms*. Society for Industrial and Applied Mathematics, Dec. 2013. DOI: 10.1137/1.9781611973402.28.

[13] Daniel Bienstock and Abhinav Verma. "Strong NP-hardness of AC power flows feasibility". *Operations Research Letters* 47.6 (Nov. 2019), pp. 494–501. DOI: 10.1016/j.orl.2019.08.009.

[14] Dan Bienstock et al. "Mathematical programming formulations for the alternating current optimal power flow problem". *4OR* 18.3 (Sept. 2020), pp. 249–292. DOI: 10.1007/s10288-020-00455-w.

[15] Christian Bingane, Miguel F. Anjos, and Sebastien Le Digabel. "Tight-and-Cheap Conic Relaxation for the AC Optimal Power Flow Problem". *IEEE Transactions on Power Systems* 33.6 (Nov. 2018), pp. 7181–7188. DOI: 10.1109/tpwrs.2018.2848965.

[16] Endre Boros and Peter L. Hammer. "Pseudo-Boolean optimization". *Discrete Applied Mathematics* 123.1-3 (Nov. 2002), pp. 155–225. DOI: 10.1016/s0166-218x(01)00341-9.

[17] Subhonmesh Bose et al. "Quadratically Constrained Quadratic Programs on Acyclic Graphs With Application to Power Flow". *IEEE Transactions on Control of Network Systems* 2.3 (Sept. 2015), pp. 278–287. DOI: 10.1109/tcns.2015.2401172.

[18] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, Mar. 2004. DOI: 10.1017/cbo9780511804441.

[19] Samuel Burer. "A gentle, geometric introduction to copositive optimization". *Mathematical Programming* 151.1 (Mar. 2015), pp. 89–116. DOI: 10.1007/s10107-015-0888-z.

[20] Samuel Burer and Kurt M. Anstreicher. "Second-order-cone constraints for extended trust-region subproblems". *SIAM Journal on Optimization* 23.1 (2013), pp. 432–451. DOI: 10.1137/110826862.

[21] Samuel Burer and Hongbo Dong. "Representing quadratically constrained quadratic programs as generalized copositive programs". *Operations Research Letters* 40.3 (May 2012), pp. 203–206. DOI: 10.1016/j.orl.2012.02.001.

[22] Mary B Cain, Richard P O'neill, Anya Castillo, et al. *History of optimal power flow and formulations*. Staff Paper. Federal Energy Regulatory Commission, 2012, pp. 1–36.

[23] Chen Chen, Alper Atamtürk, and Shmuel S. Oren. "A spatial branch-and-cut method for nonconvex QCQP with bounded complex variables". *Mathematical Programming* 165.2 (Dec. 2016), pp. 549–577. DOI: 10.1007/s10107-016-1095-2.

[24] Carleton Coffrin, Hassan L. Hijazi, and Pascal Van Hentenryck. "The QC Relaxation: A Theoretical and Computational Study on Optimal Power Flow". *IEEE Transactions on Power Systems* 31.4 (July 2016), pp. 3008–3018. DOI: 10.1109/tpwrs.2015.2463111.

[25] Samuel Coogan et al. "Offset optimization in signalized traffic networks via semidefinite relaxation". *Transportation Research Part B: Methodological* 100 (June 2017), pp. 82–92. DOI: 10.1016/j.trb.2017.01.016.

[26] CVX Research, Inc. *CVX: Matlab Software for Disciplined Convex Programming, version 2.1*. http://cvxr.com/cvx. Dec. 2018.

[27] George B. Dantzig. *Linear Programming and Extensions*. Princeton University Press, Princeton, NJ, 1963.

[28] Michel Deza and Monique Laurent. "Applications of cut polyhedra — I". *Journal of Computational and Applied Mathematics* 55.2 (Nov. 1994), pp. 191–216. DOI: 10.1016/0377-0427(94)90020-5.

[29] Michel Deza and Monique Laurent. "Applications of cut polyhedra — II". *Journal of Computational and Applied Mathematics* 55.2 (Nov. 1994), pp. 217–247. DOI: 10.1016/0377-0427(94)90021-3.

[30] Mirjam Dür. "Copositive Programming – a Survey". In: *Recent Advances in Optimization and its Applications in Engineering*. Springer Berlin Heidelberg, 2010, pp. 3–20. DOI: 10.1007/978-3-642-12598-0_1.

[31] Anders Eltved and Martin S. Andersen. "Sufficient Conditions for Exact Semidefinite Relaxation of Homogeneous Quadratically Constrained Quadratic Programs with Forest Structure". Submitted. 2020.

[32] Anders Eltved and Samuel Burer. "Strengthened SDP Relaxation for an Extended Trust Region Subproblem with an Application to Optimal Power Flow". *arXiv e-prints*, arXiv:2009.12704 (Sept. 2020), arXiv:2009.12704. arXiv: 2009.12704 [math.OC].

[33] Anders Eltved, Joachim Dahl, and Martin S. Andersen. "On the robustness and scalability of semidefinite relaxation for optimal power flow problems". *Optimization and Engineering* 21.2 (Mar. 2019), pp. 375–392. DOI: 10.1007/s11081-019-09427-4.

[34] Tetsuya Fujie and Masakazu Kojima. "Semidefinite Programming Relaxation for Nonconvex Quadratic Programs". *Journal of Global Optimization* 10.4 (June 1997), pp. 367–380. DOI: 10.1023/A:1008282830093.

[35] Mituhiro Fukuda et al. "Exploiting Sparsity in Semidefinite Programming via Matrix Completion I: General Framework". *SIAM Journal on Optimization* 11.3 (Jan. 2001), pp. 647–674. DOI: 10.1137/s1052623400366218.

[36] Lingwen Gan et al. "Exact Convex Relaxation of Optimal Power Flow in Radial Networks". *IEEE Transactions on Automatic Control* 60.1 (Jan. 2015), pp. 72–87. DOI: 10.1109/tac.2014.2332712.

[37] A. M. Geoffrion. "Lagrangean relaxation for integer programming". In: *Approaches to Integer Programming*. Ed. by M. L. Balinski. Berlin, Heidelberg: Springer Berlin Heidelberg, 1974, pp. 82–114. DOI: 10.1007/BFb0120690.

[38] J. Duncan Glover, Mulukutla S. Sarma, and Thomas Overbye. *Power System Analysis & Design, SI Version*. Cengage Learning, Aug. 2012.

[39] Michel X. Goemans and David P. Williamson. "Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming". *Journal of the ACM* 42.6 (Nov. 1995), pp. 1115–1145. DOI: 10.1145/227683.227684.

[40] S. Gopinath et al. *Proving Global Optimality of ACOPF Solutions*. 2020. arXiv: 1910.03716 [math.OC].

[41] LLC Gurobi Optimization. *Gurobi Optimizer Reference Manual*. 2020.

[42] Christoph Helmberg. *Semidefinite Programming for Combinatorial Optimization*. 2000.

[43] Christoph Helmberg and Franz Rendl. "Solving quadratic (0, 1)-problems by semidefinite programs and cutting planes". *Mathematical Programming* 82.3 (Aug. 1998), pp. 291–315. DOI: 10.1007/bf01580072.

[44] Roger A. Horn and Charles R. Johnson. *Matrix analysis*. Cambridge university press, 2012.

[45] Yongwei Huang and Daniel P. Palomar. "Randomized Algorithms for Optimal Solutions of Double-Sided QCQP With Applications in Signal Processing". *IEEE Transactions on Signal Processing* 62.5 (Mar. 2014), pp. 1093–1108. DOI: 10.1109/tsp.2013.2297683.

[46] Yongwei Huang and Shuzhong Zhang. "Complex Matrix Decomposition and Quadratic Programming". *Mathematics of Operations Research* 32.3 (Aug. 2007), pp. 758–768. DOI: 10.1287/moor.1070.0268.

[47] R. A. Jabr. "Radial distribution load flow using conic programming". 21.3 (2006), pp. 1458–1459. DOI: 10.1109/TPWRS.2006.879234.

[48] Rujun Jiang and Duan Li. "Second order cone constrained convex relaxations for nonconvex quadratically constrained quadratic programming". *Journal of Global Optimization* 75.2 (June 2019), pp. 461–494. DOI: 10.1007/s10898-019-00793-y.

[49] Charles R. Johnson et al. "On the relative position of multiple eigenvalues in the spectrum of an Hermitian matrix with a given graph". *Linear Algebra and its Applications* 363 (Apr. 2003), pp. 147–159. DOI: 10.1016/s0024-3795(01)00589-4.

[50] Faiz A. Al-Khayyal, Christian Larsen, and Timothy Van Voorhis. "A relaxation method for nonconvex quadratically constrained quadratic programs". *Journal of Global Optimization* 6.3 (Apr. 1995), pp. 215–230. DOI: 10.1007/bf01099462.

[51] Burak Kocuk, Santanu S. Dey, and X. Andy Sun. "Strong SOCP Relaxations for the Optimal Power Flow Problem". *Operations Research* 64.6 (Dec. 2016), pp. 1177–1196. DOI: 10.1287/opre.2016.1489.

[52] Jean B. Lasserre. "A Sum of Squares Approximation of Nonnegative Polynomials". *SIAM Review* 49.4 (Jan. 2007), pp. 651–669. DOI: 10.1137/070693709.

[53] Monique Laurent. "A Comparison of the Sherali-Adams, Lovász-Schrijver, and Lasserre Relaxations for 0–1 Programming". *Mathematics of Operations Research* 28.3 (Aug. 2003), pp. 470–496. DOI: 10.1287/moor.28.3.470.16391.

[54] Monique Laurent. "Sums of Squares, Moment Matrices and Optimization Over Polynomials". In: *Emerging Applications of Algebraic Geometry*. Springer New York, Sept. 2008, pp. 157–270. DOI: 10.1007/978-0-387-09686-5_7.

[55] Javad Lavaei and Steven H. Low. "Zero Duality Gap in Optimal Power Flow Problem". *IEEE Transactions on Power Systems* 27.1 (Feb. 2012), pp. 92–107. DOI: 10.1109/tpwrs.2011.2160974.

[56] Karsten Lehmann, Alban Grastien, and Pascal Van Hentenryck. "AC-Feasibility on Tree Networks is NP-Hard". *IEEE Transactions on Power Systems* 31.1 (Jan. 2016), pp. 798–801. DOI: 10.1109/tpwrs.2015.2407363.

[57] Alex Lemon, Anthony Man-Cho So, and Yinyu Ye. "Low-Rank Semidefinite Programming: Theory and Applications". *Foundations and Trends® in Optimization* 2.1-2 (2016), pp. 1–156. DOI: 10.1561/2400000009.

[58] J. Lofberg. "YALMIP : a toolbox for modeling and optimization in MATLAB". In: *2004 IEEE International Conference on Robotics and Automation (IEEE Cat. No.04CH37508)*. 2004, pp. 284–289. DOI: 10.1109/CACSD.2004.1393890.

[59] László Lovász. "On the Shannon capacity of a graph". *IEEE Transactions on Information Theory* 25.1 (Jan. 1979), pp. 1–7. DOI: 10.1109/tit.1979.1055985.

[60] László Lovász and Alexander Schrijver. "Cones of matrices and set-functions and 0–1 optimization". *SIAM journal on optimization* 1.2 (1991), pp. 166–190.

[61] Steven H. Low. "Convex Relaxation of Optimal Power Flow—Part I: Formulations and Equivalence". *IEEE Transactions on Control of Network Systems* 1.1 (Mar. 2014), pp. 15–27. DOI: 10.1109/tcns.2014.2309732.

[62] Steven H. Low. "Convex Relaxation of Optimal Power Flow—Part II: Exactness". *IEEE Transactions on Control of Network Systems* 1.2 (June 2014), pp. 177–189. DOI: 10.1109/tcns.2014.2323634.

[63] Zhi-Quan Luo and Wei Yu. "An introduction to convex optimization for communications and signal processing". *IEEE Journal on Selected Areas in Communications* 24.8 (Aug. 2006), pp. 1426–1438. DOI: 10.1109/jsac.2006.879347.

[64] Zhi-Quan Luo et al. "Semidefinite Relaxation of Quadratic Optimization Problems". *IEEE Signal Processing Magazine* 27.3 (May 2010), pp. 20–34. DOI: 10.1109/msp.2010.936019.

[65] Garth P. McCormick. "Computability of global solutions to factorable nonconvex programs: Part I — Convex underestimating problems". *Mathematical Programming* 10.1 (Dec. 1976), pp. 147–175. DOI: 10.1007/bf01580665.

[66] Amin Mobasher et al. "A Near-Maximum-Likelihood Decoding Algorithm for MIMO Systems Based on Semi-Definite Programming". *IEEE Transactions on Information Theory* 53.11 (Nov. 2007), pp. 3869–3886. DOI: 10.1109/tit.2007.907472.

[67] Daniel K. Molzahn and Ian A. Hiskens. "Moment-based relaxation of the optimal power flow problem". In: *2014 Power Systems Computation Conference*. IEEE, Aug. 2014. DOI: 10.1109/pscc.2014.7038397.

[68] Daniel K. Molzahn and Ian A. Hiskens. "A Survey of Relaxations and Approximations of the Power Flow Equations". *Foundations and Trends® in Electric Energy Systems* 4.1-2 (2019), pp. 1–221. DOI: 10.1561/3100000012.

[69] MOSEK ApS. *The MOSEK optimization toolbox for MATLAB manual. Version 9.0.105*. 2019.

[70] Yuri Nesterov, Henry Wolkowicz, and Yinyu Ye. "Semidefinite Programming Relaxations of Nonconvex Quadratic Optimization". In: *International Series in Operations Research & Management Science*. Springer US, 2000, pp. 361–419. DOI: 10.1007/978-1-4615-4381-7_13.

[71] Yurii Nesterov and Arkadii Nemirovskii. *Interior-point polynomial algorithms in convex programming*. SIAM, 1994.

[72] Jorge Nocedal and Stephen Wright. *Numerical Optimization*. Ed. by Thomas V. Mikosch. Springer-Verlag New York, July 2006. DOI: 10.1007/978-0-387-40065-5.

[73] Qiuyu Peng and Steven H. Low. "Distributed Optimal Power Flow Algorithm for Radial Networks, I: Balanced Single Phase Case". *IEEE Transactions on Smart Grid* 9.1 (Jan. 2018), pp. 111–121. DOI: 10.1109/tsg.2016.2546305.

[74] S. Poljak, F. Rendl, and H. Wolkowicz. "A recipe for semidefinite relaxation for (0, 1)-quadratic programming". *Journal of Global Optimization* 7.1 (July 1995), pp. 51–73. DOI: 10.1007/bf01100205.

[75] Svatopluk Poljak and Zsolt Tuza. "Maximum cuts and large bipartite subgraphs". In: ed. by William Cook, László Lovász, and Paul Seymour. Vol. 20. DIMACS Series in Discrete Mathematics and Theoretical Computer Science. American Mathematical Society, 1995, pp. 181–244.

[76] Hanif D. Sherali and Warren P. Adams. *A reformulation-linearization technique for solving discrete and continuous nonconvex problems*. Vol. 31. Nonconvex Optimization and its Applications. Kluwer Academic Publishers, Dordrecht, 1999, pp. xxiv+514. DOI: 10.1007/978-1-4757-4388-3.

[77] Naum Zuselevich Shor. "Quadratic optimization problems". *Soviet Journal of Computer and Systems Sciences* 25 (1987), pp. 1–11.

[78] Naum Zuselevich Shor. "Dual quadratic estimates in polynomial and Boolean programming". *Annals of Operations Research* 25.1 (1990), pp. 163–168.

[79] Jos F. Sturm. "Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones". *Optimization Methods and Software* 11.1 (1999), pp. 625–653. DOI: 10.1080/10556789908805766.

[80] Jos F. Sturm and Shuzhong Zhang. "On Cones of Nonnegative Quadratic Functions". *Mathematics of Operations Research* 28.2 (May 2003), pp. 246–267. DOI: 10.1287/moor.28.2.246.14485.

[81] Joshua A. Taylor. *Convex Optimization of Power Systems*. Cambridge University Press, 2015.

[82] K. C. Toh, M. J. Todd, and R. H. Tutuncu. "SDPT3 — a Matlab software package for semidefinite programming". *Optimization Methods and Software* 11 (1999), pp. 545–581.

[83] Lieven Vandenberghe and Martin S. Andersen. "Chordal Graphs and Semidefinite Optimization". *Foundations and Trends® in Optimization* 1.4 (2015), pp. 241–433. DOI: 10.1561/2400000006.

[84] Lieven Vandenberghe and Stephen Boyd. "Semidefinite Programming". *SIAM Review* 38.1 (Mar. 1996), pp. 49–95. DOI: 10.1137/1038003.

[85] Irène Waldspurger, Alexandre d'Aspremont, and Stéphane Mallat. "Phase recovery, MaxCut and complex semidefinite programming". *Mathematical Programming* 149.1 (Feb. 2015), pp. 47–81. DOI: 10.1007/s10107-013-0738-9.

[86] Henry Wolkowicz. "Semidefinite and Lagrangian Relaxations for Hard Combinatorial Problems". In: *System Modelling and Optimization*. Springer US, 2000, pp. 269–309. DOI: 10.1007/978-0-387-35514-6_13.

[87] Henry Wolkowicz and Miguel F. Anjos. "Semidefinite programming for discrete optimization and matrix completion problems". *Discrete Applied Mathematics* 123.1-3 (Nov. 2002), pp. 513–577. DOI: 10.1016/s0166-218x(01)00352-3.

[88] Yinyu Ye. *Interior Point Algorithms: Theory and Analysis*. Wiley Series in Discrete Mathematics and Optimization. Wiley, Aug. 1997.

[89] Yinyu Ye and Shuzhong Zhang. "New results on quadratic minimization". *SIAM Journal on Optimization* 14.1 (2003), pp. 245–267.

[90] Richard Y. Zhang and Javad Lavaei. "Sparse semidefinite programs with guaranteed near-linear time complexity via dualized clique tree conversion". *Mathematical Programming* (May 2020). DOI: 10.1007/s10107-020-01516-y.

[91] Y. Zheng et al. *CDCS: Cone Decomposition Conic Solver, version 1.1*. Sept. 2016.

[92] R. D. Zimmerman and C. E. Murillo-Sánchez. *MATPOWER 6.0 User's Manual*. Dec. 2016.

Appendix A

Paper A

Reprinted by permission from Springer Nature:

[33] Anders Eltved, Joachim Dahl, and Martin S. Andersen. "On the robustness and scalability of semidefinite relaxation for optimal power flow problems". *Optimization and Engineering* 21.2 (Mar. 2019), pp. 375–392. DOI: 10.1007/s11081-019-09427-4

Status: Published.

Noname manuscript No.
(will be inserted by the editor)

On the Robustness and Scalability of Semidefinite Relaxation for Optimal Power Flow Problems

Anders Eltved · Joachim Dahl · Martin S. Andersen

the date of receipt and acceptance should be inserted later

Abstract Semidefinite relaxation techniques have shown great promise for nonconvex optimal power flow problems. However, a number of independent numerical experiments have led to concerns about scalability and robustness of existing SDP solvers. To address these concerns, we investigate some numerical aspects of the problem and compare different state-of-the-art solvers. Our results demonstrate that semidefinite relaxations of large problem instances with on the order of 10,000 buses can be solved reliably and to reasonable accuracy within minutes. Furthermore, the semidefinite relaxation of a test case with 25,000 buses can be solved reliably within half an hour; the largest test case with 82,000 buses is solved within eight hours. We also compare the lower bound obtained via semidefinite relaxation to locally optimal solutions obtained with nonlinear optimization methods and calculate the optimality gap.

Keywords AC Optimal Power Flow · Semidefinite Relaxation · Optimization · Numerical Analysis

1 Introduction

The alternating current optimal power flow (ACOPF) problem is a nonlinear optimization problem that is concerned with finding an optimal operating point for a power system network. Today, more than 60 years after it was first studied by Carpentier (1962), the problem still receives considerable attention because of the challenging nature of the problem and its important

A. Eltved and M. S. Andersen
Department of Applied Mathematics and Computer Science, Technical University of Denmark, 2800 Kgs. Lyngby, Denmark (e-mail: {aelt,mskan}@dtu.dk)

J. Dahl
MOSEK ApS, Fruebjergvej 3, Symbion Science Park, 2100 Copenhagen, Denmark (e-mail: dahl.joachim@gmail.com)

role in power system planning and operation. Many optimization methods have been applied to the ACOPF problem, including general nonlinear optimization techniques, interior-point methods, and meta-heuristic optimization methods (Taylor, 2015).

Following the work of Jabr (2006) and Bai et al. (2008), the use of convex relaxation techniques applied to the ACOPF problem has been explored extensively; see *e.g.* (Low, 2014a,b) for a recent survey. The interest in these techniques is driven by the fact that the solution to a relaxed problem provides either a globally optimal solution to the original problem or a global lower bound that can be used to assess the quality of locally optimal solutions found by other means. Moreover, a solution to an SDR may also be used to guide a load flow study (Mak et al., 2018) in order to find a feasible operating point.

Different convex relaxations of the ACOPF problem have been proposed and studied, including a second-order cone relaxation (SOCR) (Jabr, 2006), a semidefinite relaxation (SDR) (Bai et al., 2008; Lavaei and Low, 2012), moment relaxations (Molzahn and Hiskens, 2015; Josz et al., 2015), and more recently, a quadratic convex relaxation (QCR) (Coffrin et al., 2016; Hijazi et al., 2017). The different relaxations vary in tightness and computational cost; we refer to (Coffrin et al., 2016) for a recent comparison of the SDR, QCR, and SOCR. For example, the SDR is generally tighter than the SOCR, but it is generally also more computationally demanding. One direction of research is dedicated to strengthening the SOCR; see *e.g.* (Kocuk et al., 2016). In an attempt to address the computational cost associated with the SDR, Andersen et al. (2014) and Bingane et al. (2018) have proposed simpler, weaker SDRs that are cheaper to solve than the standard SDR. The QCR is generally neither weaker nor stronger than the SDR, but it is computationally cheaper and often provides a lower bound of similar quality as that of the SDR.

The high computational cost of solving an SDR of a large ACOPF problem has given rise to concerns about robustness and scalability (Hijazi et al., 2016, 2017; Madani et al., 2017). These concerns are supported by numerical experiments that show that solving the SDR is not only much slower than other approaches, but also more unreliable (Coffrin et al., 2016). Our goal with this paper is to address concerns regarding robustness and scalability by demonstrating numerically that an SDR of the ACOPF problem can be solved both reliably and within minutes using commodity hardware, even for large networks with on the order of 10,000 buses. Our contribution is therefore confined to numerical considerations and implementation details (Section 2) as well as numerical experiments (Section 3) with the purpose of investigating scalability, accuracy, and robustness for different solvers. What differentiates our implementation from most implementations that have been described and investigated in the literature is the fact that we construct the SDR manually without the use of modeling tools such as YALMIP (Löfberg, 2004) and CVX (Grant and Boyd, 2008). Although this manual approach can be both inflexible and cumbersome, it is typically much faster and allows us to control the exact problem formulation, avoiding automatic transformations that may ad-

versely affect the size and conditioning of the SDR problem. We remark that some modeling tools allow some degree of control over the problem formulation (*e.g.*, through options), but it is generally difficult for non-expert users to predict the final problem formulation.

Notation The set $\mathcal{K}_{\mathrm{q}}^n = \{(t,x) \in \mathbb{R} \times \mathbb{R}^{n-1} \,|\, \|x\|_2 \leq t\}$ denotes the second-order cone in $\mathbb{R}^n$, $\mathbb{S}^n$ denotes the set of symmetric matrices of order n, and $\mathbb{H}^n$ is the set of Hermitian matrices of order n. The sets $\mathbb{S}_+^n$ and $\mathbb{H}_+^n$ are the cones of positive semidefinite matrices in $\mathbb{S}^n$ and $\mathbb{H}^n$, respectively. Since the symmetric matrices of order n form a vector space of dimension $n(n+1)/2$, the cone $\mathbb{S}_+^n$ can be reparameterized as $\mathcal{K}_{\mathrm{s}}^n = \{\mathbf{svec}(X) \,|\, X \in \mathbb{S}_+^n\} \subset \mathbb{R}^{n(n+1)/2}$ where $\mathbf{svec}(\cdot)$ is an injective function that maps a symmetric matrix of order n to a vector of length $n(n+1)/2$. Similarly, we define $\mathcal{K}_{\mathrm{h}}^n = \{\mathbf{hvec}(X) \,|\, X \in \mathbb{H}_+^n\} \subset \mathbb{R}^{n^2}$ where $\mathbf{hvec}(\cdot)$ maps a Hermitian matrix of order n to a vector of length n^2. The inner product between two matrices $A, B \in \mathbb{H}^n$ is $\mathbf{tr}(A^H B)$ where $\mathbf{tr}(A)$ denotes the trace of a square matrix A. Given a complex number $c = a + \jmath b$ where $\jmath = \sqrt{-1}$, $\Re(c)$ denotes the real part a, $\Im(c)$ denotes the imaginary part b, and c^* denotes the complex conjugate of c.

2 Method

2.1 The AC Optimal Power Flow Problem

An AC power system in steady state can be modeled as a directed graph where the set of nodes $\mathcal{N} = \{1, 2, \ldots, n\}$ corresponds to a set of n power buses, and the set of edges $\mathcal{L} \in \mathcal{N} \times \mathcal{N}$ corresponds to transmission lines, *i.e.*, $(k,l) \in \mathcal{L}$ if there is a line from bus k to bus l. The set $\mathcal{L}^{\mathrm{fl}} \subseteq \mathcal{L}$ consists of all transmission lines with a flow constraint, $\mathcal{L}^{\mathrm{pa}} \subseteq \mathcal{L}$ consists of all transmission lines with a phase-angle difference constraint, $\mathcal{G}_k$ denotes a (possibly empty) set of generators associated with bus k, and $\mathcal{G} = \bigcup_{k \in \mathcal{N}} \mathcal{G}_k$ is the set of all generators. The power produced by generator $g \in \mathcal{G}$ is $s_g = p_g + \jmath q_g$, and at each power bus $k \in \mathcal{N}$, we define a complex load (*i.e.*, demand) $S_k^d = P_k^d + \jmath Q_k^d$, a complex voltage v_k, and a complex current i_k. To simplify notation, we define a vector of voltages $v = (v_1, v_2, \ldots, v_n)$ and a vector of currents $i = (i_1, i_2, \ldots, i_n)$. With this notation, the ACOPF problem can be expressed as

$$\text{minimize} \quad \sum_{g \in \mathcal{G}} f_g(p_g) \tag{1a}$$

subject to

$$i_k^* v_k = \sum_{g \in \mathcal{G}_k} s_g - S_k^d, \qquad k \in \mathcal{N} \tag{1b}$$

$$P_g^{\min} \leq p_g \leq P_g^{\max}, \qquad g \in \mathcal{G} \tag{1c}$$

$$Q_g^{\min} \le q_g \le Q_g^{\max}, \qquad g \in \mathcal{G} \tag{1d}$$
$$V_k^{\min} \le |v_k| \le V_k^{\max}, \qquad k \in \mathcal{N} \tag{1e}$$
$$\left|S_{k,l}^{\mathrm{fl}}(v)\right| \le S_{k,l}^{\max}, \qquad (k,l) \in \mathcal{L}^{\mathrm{fl}} \tag{1f}$$
$$\left|S_{l,k}^{\mathrm{fl}}(v)\right| \le S_{l,k}^{\max}, \qquad (k,l) \in \mathcal{L}^{\mathrm{fl}} \tag{1g}$$
$$\phi_{k,l}^{\min} \le \angle(v_k v_l^*) \le \phi_{k,l}^{\max}, \qquad (k,l) \in \mathcal{L}^{\mathrm{pa}} \tag{1h}$$
$$i = Yv \tag{1i}$$

with variables $i \in \mathbb{C}^n$, $v \in \mathbb{C}^n$, and $s \in \mathbb{C}^{|\mathcal{G}|}$, and where $i = Yv$ corresponds to Ohm's law in matrix form, given the network admittance matrix $Y \in \mathbb{C}^{n \times n}$. The cost of generation for generator g is given by $f_g(p_g)$, and we will restrict our attention to convex quadratic generation cost functions, *i.e.*,

$$f_g(p_g) = \alpha_g p_g^2 + \beta_g p_g + \gamma_g, \tag{2}$$

where the parameters $\alpha_g \ge 0$, β_g, and γ_g are given. The constraints (1b) are power balance equations, (1c) and (1d) are generation limits, (1e) are voltage magnitude limits, (1f) and (1g) are transmission line flow constraints, and (1h) are phase-angle difference constraints. The flow from bus k to bus l is given by $S_{k,l}^{\mathrm{fl}}(v) = v^H T_{k,l} v + j v^H \widetilde{T}_{k,l} v$ (provided that $(k,l) \in \mathcal{L}^{\mathrm{fl}}$ or $(l,k) \in \mathcal{L}^{\mathrm{fl}}$) where $T_{k,l} \in \mathbb{H}^n$ and $\widetilde{T}_{k,l} \in \mathbb{H}^n$ are given.

2.2 Semidefinite Relaxation

Roughly following the steps described in (Andersen et al., 2014), we start by reformulating the ACOPF problem (1). Specifically, we perform the following steps:

1. Eliminate $i = Yv$ and substitute $P_g^{\min} + p_g^{\mathrm{l}}$ for p_g, $Q_g^{\min} + q_g^{\mathrm{l}}$ for q_g, and X for vv^H.
2. Drop constant terms in the objective:

$$\begin{aligned} f(p_g) &= \alpha_g (P_g^{\min} + p_g^{\mathrm{l}})^2 + \beta_g (P_g^{\min} + p_g^{\mathrm{l}}) + \gamma_g \\ &= \alpha_g (p_g^{\mathrm{l}})^2 + \tilde{\beta}_g p_g^{\mathrm{l}} + \text{const.} \end{aligned}$$

   where $\tilde{\beta}_g = (\beta_g + 2\alpha_g P_g^{\min})$.
3. Introduce an auxiliary variable t_g for each $g \in \mathcal{G}^{\mathrm{quad}} = \{g \in \mathcal{G} \,|\, \alpha_g > 0\}$ and include epigraph constraint

$$\alpha_g (p_g^{\mathrm{l}})^2 \le t_g \;\Leftrightarrow\; \begin{bmatrix} 1/2 + t_g \\ 1/2 - t_g \\ \sqrt{2\alpha_g} p_g^{\mathrm{l}} \end{bmatrix} \in \mathcal{K}_{\mathrm{q}}^3.$$

4. Introduce slack variables to obtain a standard-form formulation.

These steps yield the equivalent problem

$$\text{minimize} \quad \sum_{g\in\mathcal{G}} \tilde{\beta}_g p_g^{\mathrm{l}} + \sum_{g\in\mathcal{G}^{\mathrm{quad}}} t_g \tag{3a}$$

subject to

$$\begin{aligned}
\mathbf{tr}(Y_k X) &= \sum_{g\in\mathcal{G}_k} (P_g^{\min} + p_g^{\mathrm{l}}) - P_k^d, && k\in\mathcal{N} && \text{(3b)}\\
\mathbf{tr}(\widetilde{Y}_k X) &= \sum_{g\in\mathcal{G}_k} (Q_g^{\min} + q_g^{\mathrm{l}}) - Q_k^d, && k\in\mathcal{N} && \text{(3c)}\\
p_g^{\mathrm{l}} + p_g^{\mathrm{u}} &= P_g^{\max} - P_g^{\min}, && g\in\mathcal{G} && \text{(3d)}\\
q_g^{\mathrm{l}} + q_g^{\mathrm{u}} &= Q_g^{\max} - Q_g^{\min}, && g\in\mathcal{G} && \text{(3e)}\\
X_{kk} - \nu_k^{\mathrm{l}} &= (V_k^{\min})^2, && k\in\mathcal{N} && \text{(3f)}\\
X_{kk} + \nu_k^{\mathrm{u}} &= (V_k^{\max})^2, && k\in\mathcal{N} && \text{(3g)}\\
z_{k,l} &= \begin{bmatrix} S_{k,l}^{\max} \\ \mathbf{tr}(T_{k,l}X) \\ \mathbf{tr}(\widetilde{T}_{k,l}X) \end{bmatrix}, && (k,l)\in\mathcal{L}^{\mathrm{fl}} && \text{(3h)}\\
z_{l,k} &= \begin{bmatrix} S_{l,k}^{\max} \\ \mathbf{tr}(T_{l,k}X) \\ \mathbf{tr}(\widetilde{T}_{l,k}X) \end{bmatrix}, && (k,l)\in\mathcal{L}^{\mathrm{fl}} && \text{(3i)}\\
w_g &= \begin{bmatrix} 1/2 + t_g \\ 1/2 - t_g \\ \sqrt{2\alpha_g} p_g^{\mathrm{l}} \end{bmatrix}, && g\in\mathcal{G}^{\mathrm{quad}} && \text{(3j)}\\
\Im(X_{kl}) &= \tan(\phi_{k,l}^{\min})\Re(X_{kl}) + y_{k,l}^{\mathrm{l}}, && (k,l)\in\mathcal{L}^{\mathrm{pa}} && \text{(3k)}\\
\Im(X_{kl}) &= \tan(\phi_{k,l}^{\max})\Re(X_{kl}) - y_{k,l}^{\mathrm{u}}, && (k,l)\in\mathcal{L}^{\mathrm{pa}} && \text{(3l)}\\
p_g^{\mathrm{l}}, p_g^{\mathrm{u}} &\geq 0, && g\in\mathcal{G} && \text{(3m)}\\
q_g^{\mathrm{l}}, q_g^{\mathrm{u}} &\geq 0, && g\in\mathcal{G} && \text{(3n)}\\
\nu_k^{\mathrm{l}}, \nu_k^{\mathrm{u}} &\geq 0, && k\in\mathcal{N} && \text{(3o)}\\
y_{k,l}^{\mathrm{l}}, y_{k,l}^{\mathrm{u}} &\geq 0, && (k,l)\in\mathcal{L}^{\mathrm{pa}} && \text{(3p)}\\
z_{k,l}, z_{l,k} &\in \mathcal{K}_{\mathrm{q}}^3, && (k,l)\in\mathcal{L}^{\mathrm{fl}} && \text{(3q)}\\
w_g &\in \mathcal{K}_{\mathrm{q}}^3, && g\in\mathcal{G}^{\mathrm{quad}} && \text{(3r)}\\
X &= vv^H && && \text{(3s)}
\end{aligned}$$

with variables $p^{\mathrm{l}}, p^{\mathrm{u}}, q^{\mathrm{l}}, q^{\mathrm{u}} \in \mathbb{R}^{|\mathcal{G}|}$, $t \in \mathbb{R}^{|\mathcal{G}^{\mathrm{quad}}|}$, $\nu^{\mathrm{l}}, \nu^{\mathrm{u}} \in \mathbb{R}^{|\mathcal{N}|}$, $y^{\mathrm{l}}, y^{\mathrm{u}} \in \mathbb{R}^{|\mathcal{L}^{\mathrm{pa}}|}$, $z_{k,l}, z_{l,k} \in \mathcal{K}_{\mathrm{q}}^3$ for $(k,l)\in\mathcal{L}^{\mathrm{fl}}$, $w_g \in \mathcal{K}_{\mathrm{q}}^3$ for $g\in\mathcal{G}^{\mathrm{quad}}$, $X\in\mathbb{H}^n$, and $v\in\mathbb{C}^n$. Notice that the constraints (3b)-(3l) are all linear. We refer the reader to (Andersen et al., 2014) for a definition of the data matrices $Y_k, \widetilde{Y}_k, T_{k,l}$, and $\widetilde{T}_{k,l}$.

The only non-convex constraint in (3) is the rank-1 condition (3s). An SDR of (3) is readily obtained by replacing (3s) by the positive semidefiniteness constraint $X \succeq 0$. The resulting SDR is a so-called cone linear program (CLP) that can be expressed as

$$\begin{aligned} \text{minimize} \quad & c^T x \\ \text{subject to} \quad & Ax = b \\ & x \in \mathcal{K} \end{aligned} \tag{4}$$

where x is the vector of variables and the cone $\mathcal{K}$ is a Cartesian product of three types of cones, *i.e.*,

$$\mathcal{K} = \mathbb{R}^{n_l}_+ \times \underbrace{\mathcal{K}^3_{\mathrm{q}} \times \cdots \times \mathcal{K}^3_{\mathrm{q}}}_{n_q} \times \mathcal{K}^n_{\mathrm{h}}.$$

Thus, the number of variables is $N = n_l + 3n_q + n^2$ where $n_l = 4|\mathcal{G}| + |\mathcal{G}^{\mathrm{quad}}| + 2|\mathcal{N}| + 2|\mathcal{L}^{\mathrm{pa}}|$ and $n_q = 2|\mathcal{L}^{\mathrm{fl}}| + |\mathcal{G}^{\mathrm{quad}}|$, and the number of equality constraints is $M = 4|\mathcal{N}| + 2|\mathcal{G}| + 2|\mathcal{L}^{\mathrm{pa}}| + 3n_q$.

2.3 Conversion

The computational cost of solving (4) with a general-purpose interior-point method becomes prohibitively large when n is large: the cost of an interior-point iteration is at least $O(n^3)$. Fortunately, the problem (4) is generally very sparse in practice, and hence the conversion method of Fukuda et al. (2001) may be used to rewrite (4) as an equivalent CLP

$$\begin{aligned} \text{minimize} \quad & \tilde{c}^T \tilde{x} \\ \text{subject to} \quad & \tilde{A}\tilde{x} = b \\ & E\tilde{x} = 0 \\ & \tilde{x} \in \widetilde{\mathcal{K}} \end{aligned} \tag{5}$$

with

$$\widetilde{\mathcal{K}} = \mathbb{R}^{n_l}_+ \times \underbrace{\mathcal{K}^3_{\mathrm{q}} \times \cdots \times \mathcal{K}^3_{\mathrm{q}}}_{n_q} \times \mathcal{K}^{r_1}_{\mathrm{h}} \times \cdots \times \mathcal{K}^{r_m}_{\mathrm{h}}.$$

The conversion essentially decomposes the cone $\mathcal{K}^n_{\mathrm{h}}$ into a Cartesian product of a number of lower-dimensional cones $\mathcal{K}^{r_1}_{\mathrm{h}} \times \cdots \times \mathcal{K}^{r_m}_{\mathrm{h}}$ at the expense of a set of coupling constraints $E\tilde{x} = 0$. This reformulation of the problem can have a dramatic effect on the computational cost of solving the SDR of the ACOPF problem, and it effectively mitigates the $O(n^3)$ bottleneck that arises with the formulation (4). Moreover, the conversion technique often induces sparsity in the system of equations that define the search direction at each interior-point iteration, reducing the cost per iteration further if the solver can exploit this type of sparsity. The conversion technique was first applied to SDRs of the ACOPF problem by Jabr (2012).

2.4 Implementation

Before turning to our numerical experiments, we briefly outline our implementation (Andersen, 2018). The code is written in Python and performs the following steps:

1. Read case file and build the CLP (4).
2. Apply conversion method: convert (4) to (5).
3. Apply Hermitian-to-symmetric transformation: map $\mathcal{K}_{\mathrm{h}}^{r_i}$ to $\mathcal{K}_{\mathrm{s}}^{2r_i}$ for $i = 1, \ldots, m$.
4. Scale the problem data to improve conditioning.

As part of the first step, we allow some preprocessing of the data: (i) slack variables p_g (or q_g) for which $P_g^{\min} = P_g^{\max}$ (or $Q_g^{\min} = Q_g^{\max}$) may be eliminated, (ii) numerical proxies for infinity which are used to indicate the absence of limits (*e.g.*, on generation) may be truncated, and (iii) a minimum resistance of transmission lines may be enforced. The Hermitian-to-symmetric transformation is a well known trick that is only necessary because the solvers used in our experiments cannot directly handle cones of Hermitian positive semidefinite matrices; see *e.g.* (Boyd and Vandenberghe, 2004). The scaling of the problem data in step 4 is a row-scaling of the equality constraints $\tilde{A}\tilde{x} = b$ in the CLP in (5). We define a vector α with elements

$$\alpha_i = \max\{\max_j |\tilde{A}_{ij}|, |b_i|, 1\}$$

and use the equivalent, scaled constraints $\mathbf{diag}(\alpha)^{-1}\tilde{A}\tilde{x} = \mathbf{diag}(\alpha)^{-1}b$. Additionally, we scale the objective to become $\tilde{c}^T\tilde{x}/\max\{\|\tilde{c}\|_2, 1\}$. This yields an equivalent problem, and we found that for some solvers, this can reduce the computational time by roughly a factor of two; we briefly return to the topic of scaling in Section 4.

3 Results

3.1 Experiments

To investigate the robustness and scalability of our methodology, we conducted a series of numerical experiments based on a collection of test cases from MATPOWER (Zimmerman et al., 2011) (which includes a number of test cases from (Josz et al., 2016)) and Power Grid Lib (PGLib-OPF, 2018) with as many as $n = 70{,}000$ power buses; we have also included a synthetic case of the continental USA from the Electric Grid Test Case Repository (Birchfield et al., 2017) with $n = 82{,}000$ power buses. We excluded cases that are infeasible and cases with generator cost functions that are neither quadratic nor linear. For each test case, we set up a CLP formulation of the SDR and solved it using five different CLP solvers: MOSEK 8.1 (MOSEK, 2015), SeDuMi 1.3 (Sturm, 1999), SDPT3 4.0 (Toh et al., 1999), SCS 1.2.7 (O'Donoghue et al.,

2016), and CDCS 1.1 (Zheng et al., 2016). MOSEK, SeDuMi, and SDPT3 are interior-point methods whereas SCS and CDCS are first-order methods based on the alternating direction method of multipliers (ADMM).

To compare our methodology to an approach based on a modeling tool, we used SDPOPF (Molzahn et al., 2013) from "MATPOWER Extras" to set up and solve an SDR of each case. SDPOPF uses YALMIP (Löfberg, 2004) to set up the problem which is then solved numerically using one of several possible solvers: we used MOSEK in order to facilitate a fair comparison. Finally, to compare our approach to a nonlinear optimization approach, we used MATPOWER to set up and solve each case with three different interior-point methods for nonlinear optimization: MIPS (Wang et al., 2007) from MATPOWER 6.1, IPOPT 3.12.9 (Wächter and Biegler, 2006) with PARDISO 6.0 (Kourounis et al., 2018), and KNITRO 10.3.1 (Byrd et al., 2006). These are all called via MATPOWER using its default initialization—the default is sometimes referred to as "flat start" since all voltages are set to 1 p.u. and the active power generation is set to the midpoint of its bounds. When successful, these solvers return a locally optimal solution that provides an upper bound on the optimal value in contrast to the SDR that provides a lower bound.

3.2 Setup

Using the implementation described in section 2.4, we processed the problem data before setting up the SDRs. Specifically, we truncated generator bounds larger than 50 times the base MVA. We remark that SDPOPF enforces a minimum transmission line resistance of 10^{-4}; in the experiments, we do not enforce a minimum resistance in our SDR.

All experiments but those involving KNITRO were conducted on an HPC node with two Intel XeonE5-2650v4 processors (a total of 24 cores) and 240 GB memory. All experiments with KNITRO were conducted on different hardware (2.5 GHz Intel Core i5 CPU, 8 GB of memory) because of license restrictions. As a result, the KNITRO computation times that we report cannot be compared directly to those reported for the other solvers. All MATLAB-based solvers were used with MATLAB R2017b, and MOSEK was called through its Python interface in Python 3.6.3. Finally, we modified the default solver options as follows: for SeDuMi, we raised the maximum number of iterations from 150 to 250; for SCS and CDCS, we limited the number of iterations to 20,000; for CDCS, we disabled "chordalize" and used the "primal" solver since this allowed us to solve the most cases; for SCS, we used the direct solver; for SDPT3 we used a value of 400 for "smallblockdim" and changed the maximum number of iterations from 100 to 250.

3.3 Robustness

We start with an investigation of robustness. Table 1 contains a summary of return statuses for the different solvers for a total of 159 test cases. The column

Table 1 Summary of return statuses by solver.

Solver	Success	Max. iter.	Failure
MOSEK	159	0	0
SeDuMi	53	0	106
SDPT3	52	0	107
SDPOPF	128	0	31
CDCS	146	13	0
SCS	17	142	0
IPOPT	133	0	26
KNITRO	145	0	14
MIPS	116	0	43

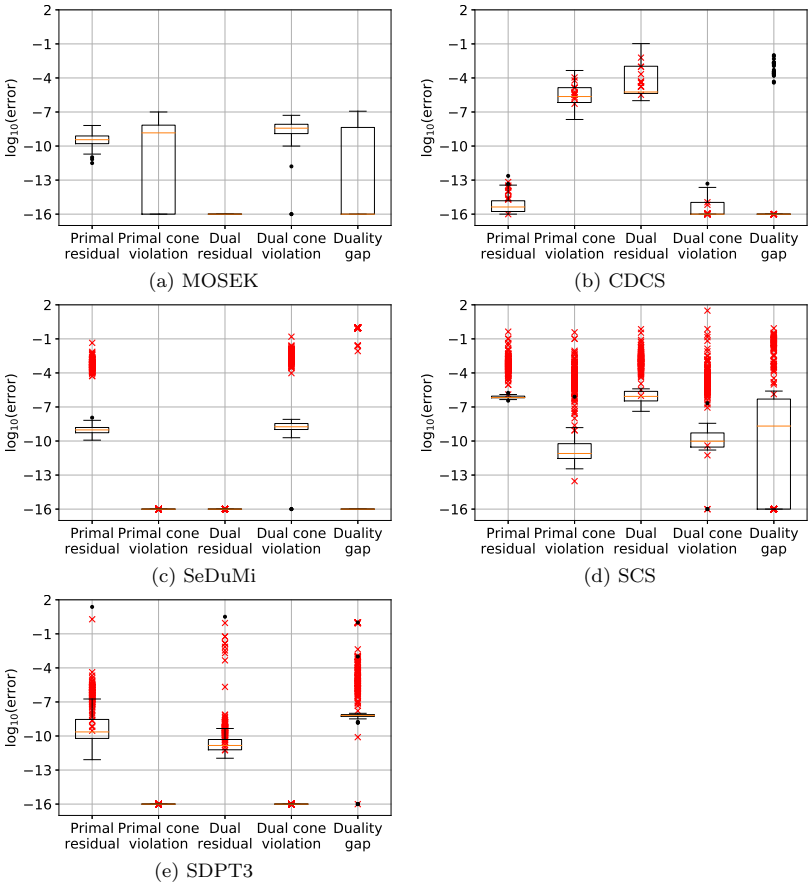
labeled "success" refers to return values that indicate successful termination with an optimal or near optimal (global or local) solution. The "failure" column refers to return values that indicate some kind of error. We remark that SDPOPF ignores phase angle constraints and fails in 31 cases because of a MATPOWER error; the solver is never called in these cases.

The results in Table 1 clearly demonstrate that the SDRs can be solved reliably using MOSEK: all cases were solved to optimality with MOSEK's default tolerances. In contrast, the nonlinear solvers IPOPT and KNITRO only succeed in roughly 85% of the cases while MIPS succeeds in approximately 75% of the cases. CDCS solves over 90% of the cases, but the accuracy and speed is poor compared to MOSEK as we show later in this section. Both SeDuMi and SCS succeed in less than 50% of the cases.

3.4 Accuracy

We now compare the solutions returned by the five CLP solvers. Since the solvers have different tolerances (*i.e.*, stopping criteria), we will compare the solvers based on the so-called "DIMACS error measures" described in (Mittelmann, 2003). Roughly speaking, these are five relative error measures quantifying the primal residual norm, primal cone violation, dual residual norm, dual cone violation, and duality gap. Fig. 1 summarizes the results in a box plot of the DIMACS measures for each solver (the smaller the error, the better).

MOSEK, shown in Fig. 1a, generally performs well with DIMACS errors below 10^{-7} in all cases. The SeDuMi errors, shown in Fig. 1c, reveal that SeDuMi returns a high-accuracy solution whenever it succeeds; the same is true for SDPT3, shown in Fig. 1e. This suggests that the default tolerances may be too strict for all but the small cases. Both CDCS and SCS generally return solutions with larger errors, as shown in Fig. 1b and 1d. This is to be expected since they are both first-order methods. While CDCS is relatively robust, it often terminates with sizable dual residuals which are indicative of low-accuracy solutions.



(a) MOSEK (b) CDCS

(c) SeDuMi (d) SCS

(e) SDPT3

Fig. 1 Box plots of logarithm of DIMACS errors. The red markers correspond to cases where the solver did not succeed. We note that in order to accommodate a logarithmic axis, we have replaced errors below 10^{-16} by this value.

3.5 Optimality Gap

Next we investigate the objective values provided by the solvers. We limit our attention to MOSEK and the nonlinear solvers IPOPT, MIPS, and KNITRO. The nonlinear solvers provide an upper bound when they terminate at a feasible point. We define the best upper bound as

$$\overline{f} = \min(f_{\text{IPOPT}}, f_{\text{KNITRO}}, f_{\text{MIPS}}), \tag{6}$$

i.e., the minimum of the objective values provided by the three solvers (if a solver does not succeed, we define its objective value to be ∞). Similarly, the

SDR (MOSEK) provides a lower bound which we denote by $\underline{f} = f_{\text{MOSEK}}$. The optimality gap may then be defined as

$$\text{gap} = \frac{\overline{f} - \underline{f}}{\overline{f}} \cdot 100\%. \tag{7}$$

The gap is equal to 0 if $\underline{f} = \overline{f}$, implying that we have a globally optimal solution. On the other hand, if the gap is large, $\overline{f}$ may be a poor local minimum and/or the SDR provides a weak lower bound $\underline{f}$.

We have made four tables listing objective values and optimality gap for all cases with more than 300 buses based on their origin: table 2 contains cases from the MATPOWER library; table 3 contains cases from PGLIB in typical operating conditions; table 4 contains cases from PGLIB with small phase angle difference constraints; table 5 contains cases from PGLIB with binding thermal limit constraints. The cases are sorted by the number of buses in ascending order. Note that the optimality gap is undefined if none of the nonlinear solvers succeed. The optimality gap is close to zero in many cases and below 1% in all but a handful of cases.

3.6 Scalability

We end this section by comparing the time required by each solver to solve the test cases. Fig. 2 shows the time used by the SDP solvers compared to the number of buses in the case. To make a fair comparison, we report computation times without preprocessing, *i.e.*, only the time required by the actual solver is recorded (we briefly discuss some considerations related to preprocessing in Section 4).

MOSEK is generally the fastest. The difference between MOSEK and SDPOPF (which also uses MOSEK, but based on the problem formulation compiled by YALMIP) highlights that the formulation of the SDR may have a significant impact on the computation time as well as robustness. The striking difference between MOSEK and CDCS, both in terms of computation time and accuracy, makes it hard to justify the use of first-order methods for highly sparse problems like these.

In addition to cost function value and optimality gap, tables 2–5 list the computation times (excluding preprocessing) for MOSEK and the three solvers IPOPT, KNITRO, and MIPS. MOSEK solves the SDR of all but one case with less than 25,000 buses in less than 10 minutes; the only exception is the case 4661_sdet from PGLIB (all three operating conditions). Solving this problem takes MOSEK around 80 minutes. The longer computation time required to solve this case compared to other cases with a similar number of buses can in part be explain by looking at the chordal embedding of the network graph. The largest clique is of size 242 which is similar to the case with 82,000 buses (238) and around three times the size of all other cases with less than 25,000 buses. The case with 25,000 power buses is solved in approximately an hour

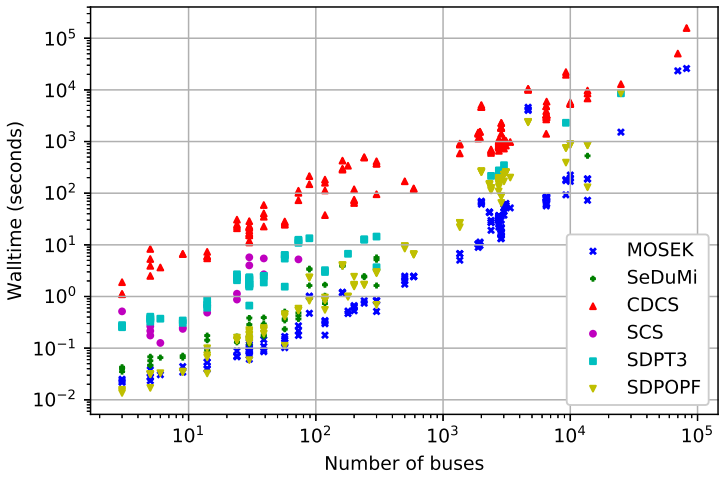


Fig. 2 Scatter plot of the time used by the SDP solvers against the number of buses for successful cases. Note that SDPOPF solves an equivalent but different SDR.

by MOSEK, and the largest cases with 70,000 and 82,000 buses are solved in around seven hours. The nonlinear solvers are typically 5-20 times faster than MOSEK (they solve a different problem!), but they sometimes fail. The RTE cases from PGLIB appear to be particularly difficult for the nonlinear solvers: in some cases, none of the nonlinear solvers succeed, and the computation times are occasionally large compared to the general trend.

4 Discussion

The difference between our formulation of the SDR and the one constructed by SDPOPF via YALMIP shows that the problem formulation can have a significant impact on computation times and robustness. Our experiments demonstrate that an SDR of the ACOPF problem can be solved accurately and reliably with the right combination of problem formulation and solver. However, it is possible that the problem formulation can be further improved. For example, as mentioned in section 2.4, the conditioning of the problem may improve with some scaling of the constraints, and this, in turn, may reduce the number of iterations and/or the computation time. We have conducted some experiments in this direction, and our preliminary results show that using MOSEK, the solution time can roughly be cut in half; the geometric mean of the speed-up obtained by means of scaling was 1.9. Indeed, the solution time for the largest test case with 25,000 buses was reduced from about one hour to half an hour with MOSEK. We did not observe a similar improvement with scaling for the other solvers. We note that SOCR and QCR implementations

could possibly benefit from scaling in a similar way. Finally, we remark that scaling may affect stopping criteria, so care must be taken when comparing the accuracy of solutions obtained with and without scaling. The QCR, proposed by Coffrin et al. (2016), provides a promising alternative to the SDR in that it is computationally cheaper and often as tight as the SDR (and in some cases even tighter). However, the findings reported in (Coffrin et al., 2016) only include SDRs of cases with less than 3,000 buses, and it is therefore unclear how the QCR and the SDR compare with respect to optimality gap for larger test cases. Moreover, the results pertaining to the SDR were obtained using an implementation based on SDPT3 and the modeling tool CVX, so the sizable gap between the two relaxations in terms of computational time will likely shrink if MOSEK and our problem formulation is used for the SDR.

The computation times reported in Section 3 did not include preprocessing time (*i.e.*, the time required to construct the SDR). To give the reader an idea of the preprocessing workload, we remark that the construction of the SDR of the case with 25 thousand buses took approximately 25 seconds or approximately 1/60 of the time required to solve the SDR with MOSEK, and the geometric average of the ratio of the solution time to the preprocessing time for cases with more than 300 buses was approximately 13, *i.e.*, preprocessing accounted for around 7% of the total time on average. In contrast, YALMIP (via SDPOPF) required approximately 6 minutes to compile the case with 25,000 buses. Comparing the ratio of the preprocessing time for YALMIP to that of our approach, we found that the geometric average was approximately 13, *i.e.*, on average it took 13 times longer with YALMIP. We note that our Python-based preprocessing code may be improved, *e.g.*, by reimplementing critical parts of the code in C. In principle, the preprocessing time may be amortized if several problem instances with the same underlying power network need to be solved. However, this would require a symbolic chordal conversion of the problem such that the problem data can easily be updated or replaced.

5 Conclusion

SDR is a promising technique that may be used to compute useful global lower bounds on the optimal value of ACOPF problems. However, concerns about robustness and scalability have cast doubt on the practical usefulness of the technique. We have shown experimentally that the problem formulation can have a significant impact on both robustness and scalability. By constructing the SDR manually instead of using a modeling tool, we avoid problem transformations that incur significant overhead. Our numerical experiments establish that SDRs of a large collection of test cases can be solved reliably with MOSEK. Moreover, the time required to solve an SDR is typically within an order of magnitude of the time required by state-of-the-art nonlinear solvers such as KNITRO and IPOPT.

References

Andersen, M. S. (2018). OPFSDR v0.2.3. https://git.io/opfsdr. Accessed on September 4 2018.

Andersen, M. S., Hansson, A., and Vandenberghe, L. (2014). Reduced-complexity semidefinite relaxations of optimal power flow problems. *IEEE Trans. Power Syst.*, 29(4):1855–1863.

Bai, X., Wei, H., Fujisawa, K., and Wang, Y. (2008). Semidefinite programming for optimal power flow problems. *International Journal of Electrical Power and Energy Systems*, 30(6-7):383–392.

Bingane, C., Anjos, M. F., and Digabel, S. L. (2018). Tight-and-cheap conic relaxation for the AC optimal power flow problem. *IEEE Trans. Power Syst.*

Birchfield, A. B., Xu, T., Gegner, K. M., Shetye, K. S., and Overbye, T. J. (2017). Grid structural characteristics as validation criteria for synthetic networks. *IEEE Transactions on Power Systems*, 32(4):3258–3265.

Boyd, S. and Vandenberghe, L. (2004). *Convex Optimization*. Cambridge University Press.

Byrd, R. H., Nocedal, J., and Waltz, R. A. (2006). Knitro: An integrated package for nonlinear optimization. In *Large Scale Nonlinear Optimization, 35–59, 2006*, pages 35–59. Springer Verlag.

Carpentier, J. (1962). Contribution á l'étude du dispatching économique. *Bulletin de la Société Française des Électriciens*, 3:431–447.

Coffrin, C., Hijazi, H., and Van Hentenryck, P. (2016). The QC relaxation: A theoretical and computational study on optimal power flow. *IEEE Trans. Power Syst.*, 31(4):3008–3018.

Fukuda, M., Kojima, M., Murota, K., and Nakata, K. (2001). Exploiting sparsity in semidefinite programming via matrix completion I: General framework. *SIAM Journal on Optimization*, 11(3):647–674.

Grant, M. and Boyd, S. (2008). Graph implementations for nonsmooth convex programs. In Blondel, V., Boyd, S., and Kimura, H., editors, *Recent Advances in Learning and Control*, Lecture Notes in Control and Information Sciences, pages 95–110. Springer-Verlag Limited.

Hijazi, H., Coffrin, C., and Hentenryck, P. V. (2017). Convex quadratic relaxations for mixed-integer nonlinear programs in power systems. *Mathematical Programming Computation*, 9(3):321–367.

Hijazi, H., Coffrin, C., and Van Hentenryck, P. (2016). Polynomial SDP cuts for optimal power flow. In *19th Power Systems Computation Conference, PSCC 2016*.

Jabr, R. A. (2006). Radial distribution load flow using conic programming. *IEEE Trans. Power Syst.*, 21(3):1458–1459.

Jabr, R. A. (2012). Exploiting sparsity in SDP relaxations of the OPF problem. *IEEE Trans. Power Syst.*, 27(2):1138–1139.

Josz, C., Fliscounakis, S., Maeght, J., and Panciatici, R. (2016). AC power flow data in MATPOWER and QCQP format: iTesla, RTE Snapshots, and PEGASE. arXiv:1603.01533v3.

Josz, C., Maeght, J., Panciatici, P., and Gilbert, J. C. (2015). Application of the moment-SOS approach to global optimization of the OPF problem. *IEEE Trans. Power Syst.*, 30(1):463–470.

Kocuk, B., Dey, S. S., and Sun, X. A. (2016). Strong socp relaxations for the optimal power flow problem. *Operations Research*, 64(6):1177–1196.

Kourounis, D., Fuchs, A., and Schenk, O. (2018). Toward the next generation of multiperiod optimal power flow solvers. *IEEE Transactions on Power Systems*, 33(4):4005–4014.

Lavaei, J. and Low, S. H. (2012). Zero duality gap in optimal power flow problem. *IEEE Trans. Power Syst.*, 27(1):92–107.

Löfberg, J. (2004). YALMIP : A toolbox for modeling and optimization in MATLAB. In *In Proceedings of the CACSD Conference*, Taipei, Taiwan.

Low, S. H. (2014a). Convex relaxation of optimal power flow—part I: Formulations and equivalence. *IEEE Transactions on Control of Network Systems*, 1(1):15–27.

Low, S. H. (2014b). Convex relaxation of optimal power flow—part II: Exactness. *IEEE Transactions on Control of Network Systems*, 1(2):177–189.

Madani, R., Kalbat, A., and Lavaei, J. (2017). A low-complexity parallelizable numerical algorithm for sparse semidefinite programming.

Mak, T. W. K., Shi, L., and Hentenryck, P. V. (2018). Phase transitions for optimality gaps in optimal power flows a study on the French transmission network. arXiv:1807.05460.

Mittelmann, H. D. (2003). An independent benchmarking of SDP and SOCP solvers. *Mathematical Programming*, 95(2):407–430.

Molzahn, D. K. and Hiskens, I. A. (2015). Sparsity-exploiting moment-based relaxations of the optimal power flow problem. *IEEE Trans. Power Syst.*, 30(6):3168–3180.

Molzahn, D. K., Holzer, J. T., Lesieutre, B. C., and DeMarco, C. L. (2013). Implementation of a large-scale optimal power flow solver based on semidefinite programming. *IEEE Trans. Power Syst.*, 28(4):3987–3998.

MOSEK (2015). *MOSEK Optimizer API for Python.*

O'Donoghue, B., Chu, E., Parikh, N., and Boyd, S. (2016). Conic optimization via operator splitting and homogeneous self-dual embedding. *Journal of Optimization Theory and Applications*, 169(3):1042–1068.

PGLib-OPF (2018). Power Grid Lib - Optimal Power Flow v18.08. https://git.io/pglib-opf. Accessed on September 4 2018.

Sturm, J. F. (1999). Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. *Optimization Methods and Software*, 11(1):625–653.

Taylor, J. A. (2015). *Convex Optimization of Power Systems.* Cambridge University Press.

Toh, K. C., Todd, M. J., and Tütüncü, R. H. (1999). SDPT3 — a Matlab software package for semidefinite programming, version 1.3. *Optimization Methods and Software*, 11(1-4):545–581.

Wächter, A. and Biegler, L. T. (2006). On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical Programming*, 106(1):25–57.

Wang, H., Murillo-Sanchez, C. E., Zimmerman, R. D., and Thomas, R. J. (2007). On computational issues of market-based optimal power flow. *IEEE Trans. Power Syst.*, 22(3):1185–1193.

Zheng, Y., Fantuzzi, G., Papachristodoulou, A., Goulart, P., and Wynn, A. (2016). CDCS: Cone Decomposition Conic Solver, version 1.1.

Zimmerman, R. D., Murillo-Sánchez, C. E., and Thomas, R. J. (2011). MATPOWER: Steady-state operations, planning, and analysis tools for power systems research and education. *IEEE Trans. Power Syst.*, 26(1):12–19.

Table 2 Cost, gap, and computation time for MATPOWER cases with more than 300 buses and without dispatchable loads. Failures are reported as 'M' (max. iterations), 'I' (termination at infeasible point), 'N' (numerical error in solver). Times shown in red correspond to failures.

		Cost				Gap	Time (sec.)			
	Case	IPOPT	KNITRO	MIPS	MOSEK		IPOPT	KNITRO	MIPS	MOSEK
MATPOWER	ACTIVSg500	7.258e+04	7.258e+04	7.258e+04	7.105e+04	2.1%	1.6	0.7	0.6	2.0
	1354pegase	7.407e+04	7.407e+04	7.407e+04	7.406e+04	0.0%	5.3	1.7	2.3	5.0
	1888rte	F	5.980e+04	F	5.960e+04	0.3%	203	17.7	1.0	8.9
	1951rte	8.174e+04	8.174e+04	F	8.173e+04	0.0%	19.9	9.7	1.2	9.7
	ACTIVSg2000	1.228e+06	1.229e+06	1.228e+06	1.228e+06	0.0%	16.2	2.5	3.3	69.9
	2383wp	1.869e+06	1.869e+06	1.869e+06	1.861e+06	0.4%	11.1	3.2	3.2	27.0
	2736sp	1.308e+06	1.308e+06	1.308e+06	1.308e+06	0.0%	27.7	2.7	3.0	29.2
	2737sop	7.776e+05	7.776e+05	7.776e+05	7.775e+05	0.0%	11.4	2.7	2.8	32.6
	2746wop	1.208e+06	1.208e+06	1.208e+06	1.208e+06	0.0%	11.5	2.8	3.2	33.7
	2746wp	1.632e+06	1.632e+06	1.632e+06	1.632e+06	0.0%	14.8	2.8	3.1	32.3
	2848rte	F	5.302e+04	F	5.301e+04	0.0%	485	31.8	1.9	13.1
	2868rte	7.979e+04	7.979e+04	F	7.979e+04	0.0%	80.7	10.7	2.4	13.3
	2869pegase	1.340e+05	1.340e+05	1.340e+05	1.340e+05	0.0%	10.7	2.9	4.6	15.2
	3012wp	2.592e+06	2.592e+06	2.592e+06	2.588e+06	0.1%	48.5	3.3	5.0	44.2
	3120sp	2.143e+06	2.143e+06	2.143e+06	2.142e+06	0.0%	16.3	3.2	5.3	53.9
	3375wp	7.412e+06	7.412e+06	7.412e+06	7.409e+06	0.0%	15.8	3.6	5.9	51.9
	6468rte	8.683e+04	8.683e+04	F	8.682e+04	0.0%	56.3	23.5	10.7	56.7
	6470rte	9.835e+04	9.835e+04	F	9.834e+04	0.0%	297	26.9	11.7	60.4
	6495rte	1.063e+05	1.063e+05	F	1.061e+05	0.2%	198	25.0	8.7	61.7
	6515rte	1.098e+05	1.098e+05	F	1.097e+05	0.1%	279	46.6	10.4	62.8
	9241pegase	3.159e+05	3.159e+05	3.159e+05	3.158e+05	0.0%	208	131	19.8	92.8
	ACTIVSg10k	2.486e+06	2.486e+06	F	2.486e+06	-0.0%	73.2	151	42.5	170
	13659pegase	F	3.861e+05	F	3.861e+05	0.0%	1,221	72.3	3,953	72.9
	ACTIVSg25k	6.018e+06	6.018e+06	F	6.017e+06	0.0%	405	51.2	85.9	1,520
	ACTIVSg70k	1.644e+07	1.644e+07	F	1.644e+07	0.0%	896	199	140	23,343
	SyntheticUSA	F	F	F	2.017e+07	—	8,582	10,534	439	25,922

Table 3 Cost, gap, and computation time for PGLIB cases in typical operating condition with more than 300 buses and without dispatchable loads. Failures are reported as 'M' (max. iterations), 'I' (termination at infeasible point), 'N' (numerical error in solver). Times shown in red correspond to failures.

		Cost				Gap	Time (sec.)			
	Case	IPOPT	KNITRO	MIPS	MOSEK		IPOPT	KNITRO	MIPS	MOSEK
PGLIB	500_tamu	7.258e+04	7.258e+04	7.258e+04	7.105e+04	2.1%	1.8	0.6	0.5	2.3
	588_sdet	3.816e+05	3.816e+05	3.816e+05	3.798e+05	0.4%	1.4	0.8	1.1	2.4
	1354_pegase	1.364e+06	1.364e+06	1.364e+06	1.356e+06	0.6%	5.0	2.1	2.7	6.7
	1888_rte	1.640e+06	1.565e+06	F	1.538e+06	1.7%	52.0	35.6	3.9	10.2
	1951_rte	2.375e+06	F	F	2.375e+06	0.0%	39.6	113	1.0	11.0
	2000_tamu	1.228e+06	1.228e+06	1.228e+06	1.228e+06	0.0%	17.0	2.6	3.7	65.2
	2316_sdet	2.257e+06	2.257e+06	2.257e+06	2.240e+06	0.7%	8.5	3.1	4.5	43.1
	2383wp_k	1.869e+06	1.869e+06	1.869e+06	1.861e+06	0.4%	11.8	4.0	3.2	27.4
	2736sp_k	1.308e+06	1.308e+06	1.308e+06	1.308e+06	0.0%	12.8	2.8	3.2	29.8
	2737sop_k	7.776e+05	7.776e+05	7.776e+05	7.775e+05	0.0%	10.9	3.0	3.0	31.1
	2746wp_k	1.632e+06	1.632e+06	1.632e+06	1.632e+06	0.0%	14.4	3.5	3.3	34.9
	2746wop_k	1.208e+06	1.208e+06	1.208e+06	1.208e+06	0.0%	12.5	3.6	3.6	36.7
	2848_rte	1.385e+06	1.385e+06	F	1.384e+06	0.0%	75.4	12.4	13.9	16.4
	2853_sdet	F	2.469e+06	2.469e+06	2.456e+06	0.5%	46.9	4.7	6.8	29.8
	2868_rte	2.260e+06	2.260e+06	F	2.260e+06	-0.0%	43.1	18.7	18.0	17.4
	2869_pegase	2.605e+06	2.605e+06	2.605e+06	2.603e+06	0.1%	20.8	5.2	6.9	20.7
	3012wp_k	2.601e+06	2.601e+06	2.601e+06	2.597e+06	0.1%	21.0	4.0	5.4	50.7
	3120sp_k	2.146e+06	2.146e+06	2.146e+06	2.145e+06	0.0%	17.3	4.0	5.7	58.4
	4661_sdet	F	2.786e+06	F	2.768e+06	0.6%	554	8.9	19.7	4,184
	6468_rte	F	2.262e+06	F	2.252e+06	0.5%	1,476	76.5	3.5	82.5
	6470_rte	F	F	F	2.545e+06	—	1,434	31.5	7.6	77.7
	6495_rte	3.478e+06	F	F	2.966e+06	14.7%	489	153	44.3	77.1
	6515_rte	F	3.197e+06	F	2.992e+06	6.4%	1,403	61.4	39.3	84.0
	9241_pegase	6.775e+06	6.775e+06	F	6.770e+06	0.1%	399	27.9	75.4	175
	10000_tamu	2.486e+06	2.486e+06	F	2.486e+06	-0.0%	98.7	113	40.1	195
	13659_pegase	1.078e+07	1.078e+07	1.078e+07	1.078e+07	0.0%	250	66.5	54.6	190

Table 4 Cost, gap, and computation time for PGLIB cases with small angle differences with more than 300 buses and without dispatchable loads. Failures are reported as 'M' (max. iterations), 'I' (termination at infeasible point), 'N' (numerical error in solver). Times shown in red correspond to failures.

		Cost				Gap	Time (sec.)			
	Case	IPOPT	KNITRO	MIPS	MOSEK		IPOPT	KNITRO	MIPS	MOSEK
PGLIB SAD	500_tamu	7.923e+04	7.923e+04	7.923e+04	7.322e+04	7.6%	2.7	0.7	0.6	2.5
	588_sdet	4.043e+05	4.043e+05	4.043e+05	3.814e+05	5.6%	2.1	0.8	1.2	2.5
	1354_pegase	1.365e+06	1.365e+06	1.365e+06	1.357e+06	0.6%	5.8	2.3	2.7	6.7
	1888_rte	F	1.640e+06	F	1.538e+06	6.2%	331	17.3	3.4	10.4
	1951_rte	2.383e+06	F	F	2.376e+06	0.3%	54.5	43.9	1.1	11.3
	2000_tamu	1.230e+06	1.230e+06	1.230e+06	1.229e+06	0.1%	31.2	3.2	3.9	61.4
	2316_sdet	2.257e+06	2.257e+06	2.257e+06	2.240e+06	0.7%	9.2	3.8	4.5	42.1
	2383wp_k	1.916e+06	1.916e+06	1.916e+06	1.905e+06	0.6%	15.8	3.9	3.4	29.8
	2736sp_k	1.329e+06	1.329e+06	1.329e+06	1.325e+06	0.4%	15.7	3.8	4.0	33.2
	2737sop_k	7.927e+05	7.927e+05	7.927e+05	7.859e+05	0.9%	15.8	4.2	3.8	33.4
	2746wp_k	1.667e+06	1.667e+06	1.667e+06	1.661e+06	0.4%	15.8	4.4	3.7	35.8
	2746wop_k	1.234e+06	1.234e+06	1.234e+06	1.226e+06	0.7%	16.7	3.9	3.8	36.9
	2848_rte	F	F	F	1.385e+06	—	558	106	13.0	16.4
	2853_sdet	F	2.495e+06	2.495e+06	2.458e+06	1.5%	87.5	5.4	6.5	30.7
	2868_rte	F	F	F	2.264e+06	—	41.3	21.7	2.7	17.4
	2869_pegase	2.620e+06	2.620e+06	2.620e+06	2.616e+06	0.2%	22.3	6.2	8.1	21.8
	3012wp_k	2.621e+06	2.621e+06	2.621e+06	2.610e+06	0.4%	21.9	5.2	6.0	52.7
	3120sp_k	2.176e+06	2.176e+06	2.176e+06	2.165e+06	0.5%	21.8	5.8	6.0	63.4
	4661_sdet	F	2.802e+06	2.802e+06	2.781e+06	0.7%	459	8.8	19.4	4,005
	6468_rte	2.262e+06	2.262e+06	F	2.252e+06	0.5%	434	45.2	3.5	80.9
	6470_rte	F	F	F	2.548e+06	—	2,637	1,519	16.1	78.6
	6495_rte	F	3.478e+06	F	2.966e+06	14.7%	2,458	54.2	22.7	82.0
	6515_rte	F	F	F	2.992e+06	—	1,234	143	6.1	79.3
	9241_pegase	6.920e+06	6.920e+06	F	6.823e+06	1.4%	142	29.1	89.5	183
	10000_tamu	F	2.486e+06	F	2.486e+06	-0.0%	1,370	94.6	38.9	196
	13659_pegase	1.090e+07	1.090e+07	1.090e+07	1.082e+07	0.7%	187	42.7	55.5	188

Table 5 Cost, gap, and computation time for heavily loaded PGLIB cases (i.e., binding thermal limits) with more than 300 buses and without dispatchable loads. Failures are reported as 'M' (max. iterations), 'I' (termination at infeasible point), 'N' (numerical error in solver). Times shown in red correspond to failures.

	Case	Cost				Gap	Time (sec.)			
		IPOPT	KNITRO	MIPS	MOSEK		IPOPT	KNITRO	MIPS	MOSEK
PGLIB API	500_tamu	4.034e+04	4.034e+04	4.034e+04	4.034e+04	-0.0%	1.1	0.6	0.5	1.7
	588_sdet	4.996e+05	4.996e+05	F	4.983e+05	0.3%	1.4	0.8	0.4	2.4
	1888_rte	F	2.262e+06	F	2.259e+06	0.2%	672	8.0	4.1	10.6
	2000_tamu	1.288e+06	1.288e+06	1.288e+06	1.275e+06	1.0%	25.7	11.8	4.2	67.6
	2316_sdet	2.774e+06	2.774e+06	2.774e+06	2.758e+06	0.6%	9.1	3.8	4.1	43.2
	2383wp_k	2.791e+05	2.791e+05	2.791e+05	2.791e+05	0.0%	7.0	1.9	1.7	19.2
	2736sp_k	6.260e+05	6.260e+05	6.260e+05	6.097e+05	2.6%	14.7	4.3	3.5	30.7
	2737sop_k	3.587e+05	3.587e+05	3.587e+05	3.485e+05	2.8%	12.9	4.0	3.1	30.6
	2746wp_k	5.818e+05	5.818e+05	5.818e+05	5.818e+05	0.0%	7.6	2.6	2.2	22.1
	2746wop_k	5.117e+05	5.117e+05	5.117e+05	5.117e+05	0.0%	6.9	2.3	1.7	24.8
	3012wp_k	7.289e+05	7.289e+05	7.289e+05	7.289e+05	0.0%	10.8	2.6	5.2	37.5
	3120sp_k	F	9.696e+05	9.696e+05	8.818e+05	9.1%	43.8	5.3	5.8	57.5
	4661_sdet	F	3.343e+06	F	3.319e+06	0.7%	207	9.7	5.7	4,604
	6468_rte	F	F	F	2.718e+06	—	1,543	172	13.2	84.2
	6470_rte	F	F	F	3.174e+06	—	1,319	859	10.0	76.1
	6495_rte	F	F	F	3.735e+06	—	2,045	111	8.5	82.1
	6515_rte	F	F	F	3.657e+06	—	2,650	54.4	5.7	85.6
	10000_tamu	1.816e+06	1.816e+06	F	1.751e+06	3.6%	153	94.0	20.1	225

Appendix B

Paper B

---

[32] Anders Eltved and Samuel Burer. "Strengthened SDP Relaxation for an Extended Trust Region Subproblem with an Application to Optimal Power Flow". *arXiv e-prints*, arXiv:2009.12704 (Sept. 2020), arXiv:2009.12704. arXiv: 2009.12704 [math.OC]

Status: Submitted.

Strengthened SDP Relaxation for an Extended Trust Region Subproblem with an Application to Optimal Power Flow

Anders Eltved* Samuel Burer†

September 26, 2020

Abstract

We study an extended trust region subproblem minimizing a nonconvex function over the hollow ball $r \leq \|x\| \leq R$ intersected with a full-dimensional second order cone (SOC) constraint of the form $\|x - c\| \leq b^T x - a$. In particular, we present a class of valid cuts that improve existing semidefinite programming (SDP) relaxations and are separable in polynomial time. We connect our cuts to the literature on the optimal power flow (OPF) problem by demonstrating that previously derived cuts capturing a convex hull important for OPF are actually just special cases of our cuts. In addition, we apply our methodology to derive a new class of closed-form, locally valid, SOC cuts for nonconvex quadratic programs over the mixed polyhedral-conic set $\{x \geq 0 : \|x\| \leq 1\}$. Finally, we show computationally on randomly generated instances that our cuts are effective in further closing the gap of the strongest SDP relaxations in the literature, especially in low dimensions.

*Department of Applied Mathematics and Computer Science, Technical University of Denmark, 2800 Kgs. Lyngby, Denmark. Email: aelt@dtu.dk.

†Department of Business Analytics, University of Iowa, Iowa City, IA, 52242-1994, USA. Email: samuel-burer@uiowa.edu.

1 Introduction

The classical *trust region subproblem (TRS)* minimizes an arbitrary quadratic function over the unit Euclidean ball defined by $\|x\| \leq R$ and is solvable in polynomial-time [10]. Many authors have studied variants of TRS that incorporate additional constraints. For example, [20] also imposes the lower bound $r \leq \|x\|$. We collectively refer to variants of TRS that incorporate more general constraints as the *extended TRS*. In this paper, we study the following specific form of the extended TRS, which incorporates the lower bound r as well as an additional SOC (second-order cone) constraint, whose "geometry" matches the ball in the sense that its Hessian is also the identity matrix:

$$\begin{aligned} \min \quad & x^T H x + 2\, g^T x && \text{(1a)} \\ \text{s.t.} \quad & r \leq \|x\| \leq R && \text{(1b)} \\ & \|x - c\| \leq b^T x - a && \text{(1c)} \end{aligned}$$

where $x \in \mathbb{R}^n$, $H = H^T \in \mathbb{R}^{n \times n}$, $g, c, b \in \mathbb{R}^n$, $a \in \mathbb{R}$, and $r, R \in \mathbb{R}_+$. Note that H is symmetric without loss of generality and that we have *not* scaled the problem to the unit ball (i.e., we do not assume $R = 1$) as is common in the TRS literature. The general upper bound R will be convenient for our presentation, especially in Section 3. The algorithm of Bienstock [3] solves (1) in polynomial time since it can be written as a nonconvex quadratic program with a fixed number of quadratic/linear constraints (in this case, four), one of which is strictly convex. However, in this paper, we are interested in developing tight convex relaxations of (1). In particular, as far as we are aware, (1) has no known tight convex relaxation.

Problem (1) includes, for example, the *two trust region subproblem*—also called the *Celis-Dennis-Tapia subproblem* [8]—in which a second ball (or ellipsoidal) constraint is added to TRS. In this case, $r = 0$, $b = 0$, and $a < 0$. Here, however, we are interested in the more general structure represented by (1c), which arises, for example, in the *optimal power flow problem (OPF)* as discussed in Section 3. More generally, the study of (1) sheds light on any nonconvex quadratically constrained quadratic program that includes a ball constraint and a second SOC constraint with identity Hessian. In Section 3, we will also show how this

structure is relevant for the mixed polyhedral-SOC set $\{x \geq 0 : \|x\| \leq R\}$. (In the concluding Section 6, we briefly mention an extension for handling different Hessians.)

Since (1) is a nonconvex problem, a standard approach is to approximate (1) by its so-called *Shor semidefinite programming (SDP) relaxation* [19], which is solvable in polynomial time:

$$\begin{aligned}
\min \quad & H \bullet X + 2\, g^T x && \text{(2a)}\\
\text{s.t.} \quad & r^2 \leq \operatorname{tr}(X) \leq R^2 && \text{(2b)}\\
& \operatorname{tr}(X) - 2\, c^T x + c^T c \leq b b^T \bullet X - 2\, a\, b^T x + a^2 && \text{(2c)}\\
& 0 \leq b^T x - a && \text{(2d)}\\
& Y(x, X) \succeq 0 && \text{(2e)}
\end{aligned}$$

where $M \bullet X := \operatorname{tr}(M^T X)$ is the trace inner product for conformal matrices and

$$Y(x, X) := \begin{pmatrix} 1 & x^T \\ x & X \end{pmatrix} \tag{3}$$

is symmetric of size $(n+1) \times (n+1)$. Note that (1c) is represented as the two constraints $\|x - c\|^2 \leq (b^T x - a)^2$ and $0 \leq b^T x - a$ before lifting to (2c)–(2d). We also define

$$\mathcal{R}_{\text{shor}} := \{(x, X) : (x, X) \text{ satisfies (2b)–(2e)}\}$$

to be the feasible set of the Shor relaxation. Then (2) can be alternatively expressed as minimizing $H \bullet X + 2\, g^T x$ over $(x, X) \in \mathcal{R}_{\text{shor}}$.

Various valid inequalities can be added to (2) in order to strengthen the Shor relaxation. For example, if $v_1^T x \geq u_1$ and $v_2^T x \geq u_2$ are any two valid linear inequalities for the feasible set of (1), then the redundant quadratic constraint $(v_1^T x - u_1)(v_2^T x - u_2) \geq 0$ can be relaxed to the valid *RLT constraint* [18]:

$$v_1 v_2^T \bullet X - u_2 v_1^T x - u_1 v_2^T x + u_1 u_2 \geq 0.$$

However, since (1) does not contain explicit linear constraints, in practice one

would need to separate over valid $v_1^T x \geq u_1$ and $v_2^T x \geq u_2$ to generate violated RLT constraints, but this separation is a bilinear subproblem, which does not appear to be solvable in polynomial time.

The difficulty of separating the RLT constraints when no linear constraints are explicitly given can be circumvented in the case of (1) as follows. By multiplying a valid $v_1^T x \geq u_1$ with the ball constraint $\|x\| \leq R$, we have the redundant quadratic SOC constraint $\|(v_1^T x - u_1)x\| \leq R(v_1^T x - u_1)$, which in turn yields the valid SOC constraint

$$\|Xv_1 - u_1 x\| \leq R(v_1^T x - u_1) \tag{4}$$

in the lifted (x, X) space. In a similar manner, $v_1^T x \geq u_1$ can be combined with $\|x - c\| \leq b^T x - a$. These are known as *SOCRLT constraints* [21, 5]. In fact, each SOCRLT constraint is a compact encoding of an entire collection of RLT constraints. For example, (4) captures all of the RLT constraints corresponding to $v_1^T x \geq u_1$ fixed and $v_2^T x \geq u_2$ varying over the supporting hyperplanes of $\|x\| \leq R$. Consequently, the collections of SOCRLT and RLT constraints for (1) are equivalent,[1] but in contrast to the RLT constraints, the SOCRLT constraints can be separated in polynomial-time based on the fact that TRS is polynomial-time solvable [5].

Anstreicher [1] introduced a further generalization of the SOCRLT constraints, called a *KSOC constraint*, which is based on relaxing a valid quadratic Kronecker-product matrix inequality. Specifically, the KSOC constraint is constructed from the following observations: first, defining $\text{SOC} := \{(v_0, v) : \|v\| \leq v_0\}$ to be the second-order cone, it is well-known that

$$\begin{pmatrix} v_0 \\ v \end{pmatrix} \in \text{SOC} \iff \begin{pmatrix} v_0 & v^T \\ v & v_0 I \end{pmatrix} \succeq 0;$$

second, it is also well-known that the Kronecker product of positive semidefinite matrices is positive semidefinite. Hence, for (1) we have the valid quadratic

[1] This differs from other papers, which often define RLT constraints only for explicitly given valid linear constraints, of which (1) has none. So, for the sake of generality, we have defined the RLT constraints allowing for *implicit* valid linear constraints.

matrix inequality

$$\begin{pmatrix} R & x^T \\ x & R\,I \end{pmatrix} \otimes \begin{pmatrix} b^T x - a & x^T - c^T \\ x - c & (b^T x - a)I \end{pmatrix} \succeq 0.$$

After relaxing this inequality in the space (x, X), we obtain the convex KSOC constraint, which captures all SOCRLT constraints (and hence all RLT constraints) and is generally stronger [1], assuming the Shor constraints remain enforced.

Summarizing, defining $\mathcal{R}_{\text{rlt}}$ and $\mathcal{R}_{\text{socrlt}}$ to be the set of (x, X) satisfying all possible RLT and SOCRLT constraints, respectively, we have

$$\mathcal{R}_{\text{shor}} \cap \mathcal{R}_{\text{ksoc}} \subseteq \mathcal{R}_{\text{shor}} \cap \mathcal{R}_{\text{socrlt}} = \mathcal{R}_{\text{shor}} \cap \mathcal{R}_{\text{rlt}}$$

where $\mathcal{R}_{\text{ksoc}}$ is the set of all (x, X) satisfying the KSOC constraint. Moreover, the first containment is proper in general. Hence, in this paper, we focus on improving the relaxation $\mathcal{R}_{\text{shor}} \cap \mathcal{R}_{\text{ksoc}}$. The paper [13] provides further insight into the strength of $\mathcal{R}_{\text{shor}} \cap \mathcal{R}_{\text{ksoc}}$ relative to other techniques in the literature.

Let $\mathcal{F}$ denote the feasible set of (1), i.e., the set of all $x \in \mathbb{R}^n$ satisfying (1b)–(1c). Strengthening the SDP relaxation can alternatively be expressed as determining valid inequalities that more accurately approximate the closed convex hull

$$\mathcal{G} := \overline{\text{conv}} \left\{ (x, xx^T) : x \in \mathcal{F} \right\}. \tag{5}$$

Note that $\mathcal{G}$ is compact because $\mathcal{F}$ is. Moreover, because linear optimization over a compact set is guaranteed to attain its optimal value at an extreme point, solving (1) amounts to optimizing the linear function $H \bullet X + 2\,g^T x$ over $\mathcal{G}$. While an exact representation of $\mathcal{G}$ is unknown, there are several closely related cases in which $\mathcal{G}$ can be described exactly; see [7, 2].

In this paper, we propose a new class of valid linear inequalities for (1) in the space (x, X), which in general strengthen $\mathcal{R}_{\text{shor}} \cap \mathcal{R}_{\text{ksoc}}$ towards $\mathcal{G}$. Each inequality is derived from several ingredients that exploit the structure of $\mathcal{F}$: the self-duality of SOC; the RLT-type valid inequality $(R - \|x\|)(\|x\| - r) \geq 0$; and knowledge of a quadratic function $q(x)$ and a linear function $l(x)$, each of

which is nonnegative over all $x \in \mathcal{F}$. We combine these ingredients to derive a valid quartic inequality, which is then relaxed to a valid quadratic inequality, which in turn yields a new valid linear inequality in (x, X).

As a small illustrative example, consider when $c = 0$ and $r = 0$, in which case $\mathcal{F}$ is defined by $\|x\| \leq R$ and $\|x\| \leq b^T x - a$. For the specific choices $q(x) = 0$ and $l(x) = 1$, our new inequality can also be derived from the following direct argument: the chain of inequalities $\|x\|^2 \leq R\|x\| \leq R(b^T x - a)$ linearizes to

$$\operatorname{tr}(X) \leq R(b^T x - a). \tag{6}$$

The following example shows that (6) is not captured by $\mathcal{R}_{\text{shor}} \cap \mathcal{R}_{\text{ksoc}}$:

Example 1. *Let* $\mathcal{F} = \{x \in \mathbb{R}^2 : \|x\| \leq 1, \|x\| \leq 1 - x_1 - x_2\}$. *Then* (6) *is* $\operatorname{tr}(X) \leq 1 - x_1 - x_2$. *Minimizing the objective* $1 - x_1 - x_2 - \operatorname{tr}(X)$ *over* $\mathcal{R}_{shor} \cap \mathcal{R}_{ksoc}$ *yields the optimal solution*

$$Y^* \approx \begin{pmatrix} 1.0000 & 0.0624 & 0.0624 \\ 0.0624 & 0.5000 & -0.3018 \\ 0.0624 & -0.3018 & 0.5000 \end{pmatrix}$$

with (approximate) optimal value -0.1248*, i.e., the optimal value is negative, which demonstrates that* (6) *is not valid for* $\mathcal{R}_{shor} \cap \mathcal{R}_{ksoc}$.

As far as we aware, inequality (6) for this special case has not yet appeared in the literature. We seek in this paper, however, an even more general procedure for deriving valid inequalities using the ingredients described in the previous paragraph.

The paper is organized as follows. In Section 2, we present the derivation of our new valid inequalities and discuss several illustrative choices of $q(x)$ and $l(x)$. We also specialize the results to $c = 0$ and $a = 0$, a case which further enables the derivation of a similar, second type of valid linear inequality in (x, X). Then, in Section 3, we show that our inequalities include those introduced in [9] for the study of the OPF problem,[2] and we extend our approach to derive a new

[2]Indeed, our initial motivation for this paper was the desire to understand the inequalities in [9] more fully.

class of valid SOC constraints for $\mathcal{G}$ when $\mathcal{F}$ equals the intersection of the ball $\|x\| \leq R$ and the nonnegative orthant. Next, in Section 4, we prove that the separation problem for our inequalities—which can be viewed as dynamically choosing the nonnegative functions $q(x)$ and $l(x)$—is polynomial-time based on the availability of any SDP relaxation in the variables (x, X), such as the relaxations $\mathcal{R}_{\text{shor}}$ or $\mathcal{R}_{\text{shor}} \cap \mathcal{R}_{\text{ksoc}}$. In this sense, we are able to "bootstrap" any existing SDP relaxation for the separation subroutine to generate valid cuts. Finally, in Section 5, we provide computational evidence that our cuts are effective in further closing the gap between (1) and $\mathcal{R}_{\text{shor}} \cap \mathcal{R}_{\text{ksoc}}$ on randomly generated problems, especially in low dimensions. We close in Section 6 with a few final thoughts and directions for future research.

This paper is accompanied by the code repository `https://github.com/A-Eltved/strengthened_sdr`, which contains full code for the paper's examples and computational results. In addition, the first author's forthcoming Ph.D. thesis [12] will contain additional discussion and extensions.

2 New Valid Inequalities

In the Introduction, we discussed the valid inequality (6) for the specific case $c = 0$ and $r = 0$. Now we assume general c and r. Analogous to (6), we use $\|x\| \leq R$ and $\|x - c\| \leq b^T x - a$ along with the self-duality of SOC to obtain the following quadratic inequality:

$$\begin{pmatrix} R \\ -x \end{pmatrix}^T \begin{pmatrix} b^T x - a \\ x - c \end{pmatrix} \geq 0 \quad \Longrightarrow \quad R(b^T x - a) \geq \operatorname{tr}(X) - c^T x. \tag{7}$$

Note that this inequality makes use of the equivalent constraint $\|-x\| \leq R$. We seek to strengthen it further by incorporating two additional ideas.

The first idea involves exploiting the lower bound $r \leq \|x\|$ and the RLT-type valid inequality $(R - \|x\|)(\|x\| - r) \geq 0$. Consider the following proposition:

Proposition 1. *Suppose $r \leq \|x\| \leq R$, and define $r\|x\|^{-2} := 0$ when $\|x\| =$*

$r = 0$. *Then*

$$\begin{pmatrix} r + R \\ \left(1 + rR\|x\|^{-2}\right) x \end{pmatrix} \in \mathit{SOC}. \tag{8}$$

Proof. If $r = 0$, then (8) reads $(R, x) \in \mathrm{SOC}$, which is true by assumption. So suppose $0 < r \leq \|x\|$. Then we wish to prove

$$(1 + rR\|x\|^{-2})\|x\| = \|x\| + rR\|x\|^{-1} \leq r + R,$$

which follows by expanding the valid expression $(R - \|x\|)(\|x\| - r) \geq 0$ and dividing by $\|x\| \geq r > 0$. □

By the proposition, analogous to (7), we have:

$$\begin{pmatrix} r + R \\ -(1 + rR\|x\|^{-2})x \end{pmatrix}^T \begin{pmatrix} b^T x - a \\ x - c \end{pmatrix} \geq 0$$
$$\iff (r + R)(b^T x - a) \geq x^T x + rR - c^T x - rR\|x\|^{-2}\, c^T x.$$

However, this inequality cannot be directly linearized in (x, X) due to the nonquadratic term $\|x\|^{-2}$. So we bound the term $r\|x\|^{-2}\, c^T x$ from above by a problem-dependent constant $[c]_{\max} \geq 0$, which satisfies $r\, c^T x \leq [c]_{\max}\, x^T x$ for all $x \in \mathcal{F}$. We then have the valid linear inequality

$$(r + R)(b^T x - a) \geq \mathrm{tr}(X) + rR - c^T x - [c]_{\max} R. \tag{9}$$

Such a $[c]_{\max}$ clearly exists. For example, $[c]_{\max} = \|c\|$ works because

$$r\, c^T x \leq r\|c\|\|x\| \leq \|c\|\|x\|^2,$$

but naturally it is advantageous to take $[c]_{\max}$ as small as possible. One method for computing a smaller $[c]_{\max} \leq \|c\|$ is binary search on $[c]_{\max}$ over the interval $[0, \|c\|]$, where at each step we check whether the optimal value of

$$\min_x \left\{ [c]_{\max}\, x^T x - r\, c^T x : \|x\| \leq R, \|x - c\| \leq b^T x - a \right\}$$

is nonnegative. The nonconvex lower bound $r \leq \|x\|$ has been excluded from

this subproblem to ensure convexity and polynomial-time solvability, which also ensures that the binary search is polynomial-time overall. Note also that, when $r = 0$ or $c = 0$, the optimal $[c]_{\max}$ equals 0.

Our second idea to improve (7) and (9) is to replace $(b^T x - a, x - c) \in \text{SOC}$ in the derivation above with another vector—but one that is still in the second-order cone. In particular, we consider the nonnegative combination

$$q_x \begin{pmatrix} R \\ x \end{pmatrix} + l_x \begin{pmatrix} b^T x - a \\ x - c \end{pmatrix} \in \text{SOC}, \tag{10}$$

where $q_x := q(x)$ is a quadratic function and $l_x := l(x)$ is a linear function, both of which are nonnegative for all $x \in \mathcal{F}$. This approach is similar to polynomial-optimization approaches such as the one pioneered in [14], which uses polynomial multipliers with limited degree to derive new, albeit redundant, constraints. Then we have the following generalization of (9):

$$\begin{pmatrix} r + R \\ -(1 + rR\|x\|^{-2})x \end{pmatrix}^T \begin{pmatrix} Rq_x + l_x(b^T x - a) \\ (q_x + l_x)x - l_x c \end{pmatrix} \geq 0$$

which rearranges and relaxes to

$$(r+R)Rq_x + (r+R)l_x(b^T x - a) \geq (q_x + l_x)\, x^T x + rR\,(q_x + l_x) - l_x c^T x - [c]_{\max}\, R\, l_x.$$

Note that the right-hand side is quartic in x, and hence this inequality cannot be directly linearized in the space (x, X). Hence, we define

$$[q + l]_{\min} := \min\{q_x + l_x : x \in \mathcal{F}\} \geq 0.$$

to get the valid quadratic inequality

$$(r+R)Rq_x + (r+R)l_x(b^T x - a) \geq [q+l]_{\min}\, x^T x + rR\,(q_x + l_x) - l_x c^T x - [c]_{\max} R\, l_x, \tag{11}$$

which can be easily linearized in (x, X) as summarized in the following theorem. Note that the theorem requires only that $[q+l]_{\min}$ be a nonnegative lower bound on the value of $q(x) + l(x)$ over $\mathcal{F}$.

Theorem 1. *Let $\mathcal{F}$ be the feasible set of (1), and let $[c]_{\max} \in [0, \|c\|]$ be given*

such that $r\,c^T x \le [c]_{\max} x^T x$ *for all* $x \in \mathcal{F}$. *In addition, let* $q(x) := x^T H_q x + 2\,g_q^T x + f_q$ *and* $l(x) := 2\,g_l^T x + f_l$ *be given such that* $q(x) \ge 0$ *and* $l(x) \ge 0$ *for all* $x \in \mathcal{F}$. *Also, let* $[q+l]_{\min} \ge 0$ *be a valid lower bound on the sum* $q(x) + l(x)$ *over all* $x \in \mathcal{F}$. *Then the linear inequality*

$$\begin{aligned}(r+R)R\left(H_q \bullet X + 2\,g_q^T x + f_q\right) + (r+R)\left(2\,g_l b^T \bullet X + (f_l b - 2a g_l)^T x - a f_l\right) \\ \ge [q+l]_{\min}\,\mathrm{tr}(X) + rR\left(H_q \bullet X + 2(g_q + g_l)^T x + (f_q + f_l)\right) \\ - \left(2\,g_l c^T \bullet X + f_l c^T x\right) - [c]_{\max} R(2\,g_l^T x + f_l) \qquad (12)\end{aligned}$$

is valid for the convex hull $\mathcal{G}$ *defined by (5).*

Note that both sides of (11) contain the term $rR\,q_x$, and so the presentation of both (11) and (12) could be simplified. However, we leave these slightly unsimplified so as to facilitate our discussion in Section 2.2 below.

Let $\hat{r}$ be any scalar in $[0, r]$. Since $\hat{r} \le \|x\|$ is also valid for $\mathcal{F}$, we can replace r by $\hat{r}$ in (12) to obtain an alternate inequality based on $\hat{r}$. In fact, considering $\hat{r}$ to be variable in this inequality while all other quantities are fixed, we see that the inequality is linear in $\hat{r}$, which implies that all such valid inequalities over $\hat{r} \in [0, r]$ are actually dominated by the two extremes $\hat{r} = 0$ and $\hat{r} = r$. We summarize this observation in the following corollary.

Corollary 1. *Under the assumptions of Theorem 1, the infinite class of inequalities gotten by replacing* r *with* $\hat{r} \in [0, r]$ *is dominated by the two inequalities (12) and*

$$\begin{aligned}R^2\left(H_q \bullet X + 2\,g_q^T x + f_q\right) + R\left(2\,g_l b^T \bullet X + (f_l b - 2a g_l)^T x - a f_l\right) \\ \ge [q+l]_{\min}\,\mathrm{tr}(X) - \left(2\,g_l c^T \bullet X + f_l c^T x\right) - [c]_{\max} R(2\,g_l^T x + f_l). \qquad (13)\end{aligned}$$

corresponding to the extremes $\hat{r} = r$ *and* $\hat{r} = 0$, *respectively.*

2.1 Example: Slab inequalities

In this subsection, we introduce a specialization of our inequalities, which we will return to in Section 3.2.

Suppose that we have knowledge of $s \in \mathbb{R}^n$ and $\lambda, \mu \in \mathbb{R}$ such that

$$\mathcal{F} \subseteq \mathcal{S} := \{x : \lambda \leq s^T x \leq \mu\}, \tag{14}$$

i.e., every $x \in \mathcal{F}$ satisfies $\lambda \leq s^T x \leq \mu$. We call $\mathcal{S}$ a valid *slab* and, abusing notation, we refer to $\mathcal{S}$ by its tuple (λ, s, μ). For example, since $\mathcal{F}$ is bounded, for any vector s with $\|s\| = 1$, choosing $\lambda = -R$ and $\mu = R$ yields a valid slab. Given any slab (λ, s, μ), we discuss two choices of nonnegative q_x and l_x.

First, define $q_x := \mu - s^T x \geq 0$ and $l_x := s^T x - \lambda \geq 0$. Note that q_x is linear in this case, and $[q + l]_{\min} = q_x + l_x = \mu - \lambda$. Then (11) becomes

$$\begin{aligned}(r + R)R(\mu - s^T x) + (r + R)(s^T x - \lambda)(b^T x - a) \\ \geq (\mu - \lambda)(x^T x + rR) - (s^T x - \lambda)c^T x - [c]_{\max} R(s^T x - \lambda).\end{aligned} \tag{15}$$

Alternatively, we could also take $q_x := s^T x - \lambda$ and $l_x := \mu - s^T x$ to obtain another, similar quadratic inequality.

Second, given the slab (λ, s, μ), we may assume without loss of generality that $\lambda + \mu \geq 0$ and $\lambda^2 \leq \mu^2$. To see this, we consider three cases. First, if both $\lambda, \mu \geq 0$, then the statement is clear. Second, if both $\lambda, \mu \leq 0$, we can use instead the equivalent representation of $\mathcal{S}$ by $-\mu \leq -s^T x \leq -\lambda$. Finally, if $\lambda < 0$ and $\mu \geq 0$ with $\lambda + \mu < 0$, then we can likewise use $(-\mu, -s, -\lambda)$ instead. Now, with $\lambda + \mu \geq 0$ and $\lambda^2 \leq \mu^2$, we then define $q_x := \mu^2 - (s^T x)^2 \geq 0$ and $l_x := (\lambda + \mu)(s^T x - \lambda) \geq 0$ so that

$$\begin{aligned} q_x + l_x &= \mu^2 - (s^T x)^2 + (\lambda + \mu)s^T x - \lambda\mu - \lambda^2 \\ &= \mu^2 + (\mu - s^T x)(s^T x - \lambda) - \lambda^2 \\ &\geq \mu^2 + 0 - \lambda^2 \geq 0. \end{aligned}$$

Hence, we obtain (11) with $[q + l]_{\min} := \mu^2 - \lambda^2 \geq 0$.

2.2 Example: Special case $c = 0$, $a = 0$, and $\lambda \geq 0$

In this subsection, we derive two cuts—see (18) below—that are closely related to the cuts just discussed in Section 2.1, and these will play a special role in Section 3.1. We assume $c = 0$ and $a = 0$, and we will use a slab (λ, s, μ) with $\lambda \geq 0$. Note that $c = 0$ implies $[c]_{\max} = 0$.

For the first cut, consider the inequality (11) with $c = 0$ and $a = 0$, which is further relaxed on the right-hand side:

$$\begin{aligned}(r + R)Rq_x + (r + R)l_x b^T x &\geq [q + l]_{\min} x^T x + rR\,(q_x + l_x) \\ &\geq [q + l]_{\min}(x^T x + rR). \end{aligned} \tag{16}$$

For the second cut, we consider a pair of functions $l_x := l(x)$ and $p_x := p(x)$ that satisfy a different relationship than the previously considered l_x and q_x. Specifically, we assume linear $l_x \geq 0$ and quadratic $p_x \geq 0$, and we require $l_x - p_x \geq 0$ for all $x \in \mathcal{F}$ as well. We also define $[l - p]_{\min} \geq 0$ to be the minimum value of $l_x - p_x$ over $\mathcal{F}$. Then we have the following result.

Proposition 2. *Suppose $c = 0$, $a = 0$, and $l_x := l(x)$ and $p_x := p(x)$ are nonnegative functions on $\mathcal{F}$ such that $l_x - p_x$ is also nonnegative on $\mathcal{F}$. Then*

$$\begin{pmatrix} l_x b^T x - r p_x \\ (l_x - p_x)x \end{pmatrix} \in \mathit{SOC}.$$

Proof. $(l_x - p_x)\|x\| = l_x\|x\| - p_x\|x\| \leq l_x b^T x - r p_x$. □

Using this proposition, the self-duality of the SOC, and Proposition 1, we have

$$\begin{pmatrix} r + R \\ -(1 + rR\|x\|^{-2})x \end{pmatrix}^T \begin{pmatrix} l_x b^T x - r p_x \\ (l_x - p_x)x \end{pmatrix} \geq 0,$$

which rearranges and relaxes to

$$\begin{aligned}(r + R)l_x b^T x - (r + R)r p_x &\geq (l_x - p_x)\, x^T x + rR\,(l_x - p_x) \\ &\geq [l - p]_{\min}(x^T x + rR). \end{aligned} \tag{17}$$

Note that (17) simplifies to $R\, l_x\, b^T x \geq [l-p]_{\min}\, x^T x$ when $r = 0$, which is a consequence of the simpler inequality $R\, b^T x \geq x^T x$; see (6) with $a = 0$. In other words, (17) appears to be interesting only when $r > 0$.

We now consider a specific choice of q_x, l_x, and p_x for the inequalities (16) and (17) based on the slab $0 \leq \lambda \leq s^T x \leq \mu$. We choose $q_x := \mu^2 - (s^T x)^2$, $l_x := (\lambda+\mu)s^T x$, and $p_x := (s^T x)^2 - \lambda^2$ as the nonnegative functions, resulting in

$$\begin{aligned} q_x + l_x &= \mu^2 - (s^T x)^2 + (\lambda+\mu)s^T x \geq \mu^2 + \lambda\mu =: [q+l]_{\min} \\ l_x - p_x &= \lambda^2 - (s^T x)^2 + (\lambda+\mu)s^T x \geq \lambda^2 + \lambda\mu =: [l-p]_{\min}, \end{aligned}$$

where the inequalities follow from the RLT inequality $(\mu - s^T x)(s^T x - \lambda) \geq 0$. Plugging these into (16)–(17), respectively, and linearizing, we obtain

$$(r+R)R(\mu^2 - ss^T \bullet X) + (r+R)(\lambda+\mu)sb^T \bullet X \geq (\mu^2+\lambda\mu)(\mathrm{tr}(X) + rR) \tag{18a}$$

$$(r+R)(\lambda+\mu)sb^T \bullet X - (r+R)r(ss^T \bullet X - \lambda^2) \geq (\lambda^2+\lambda\mu)(\mathrm{tr}(X) + rR). \tag{18b}$$

3 Applications

In this section, we explore two applications of the inequalities developed in Section 2. The first application shows that the valid inequalities for the optimal power flow problem (OPF) derived in [9] are in fact just special cases of our inequalities, whereas the derivation in [9] was specifically tailored to OPF. Our second application investigates the convex hull of $\mathcal{G}$, where—departing from the form of (1)—$\mathcal{F}$ equals the intersection of the ball with the nonnegative orthant, i.e., $\mathcal{F}$ possesses polyhedral aspects as well. We study this form of $\mathcal{F}$ since it is relevant for any bounded feasible set with nonnegative variables, where the bound is given by a Euclidean ball.

3.1 Optimal power flow problem

In this subsection, we consider a result of Chen et al. [9], which provides an exact formulation for the convex hull of a nonconvex, quadratically constrained set appearing in the study of the optimal power flow (OPF) problem. In particular, the authors added two new linear inequalities to the Shor relaxation in order to capture the convex hull. Whereas these two inequalities were specifically derived for OPF, we will show that they are just special cases of (18) derived in Section 2.2. For additional background on convex relaxations of OPF, we refer the reader to the two-part survey [15, 16].

We restate the result of Chen et al. using their notation. Let $\mathcal{J}_C \subseteq \mathbb{R}^4$ be the convex hull of the following nonconvex quadratic system:

$$L_{jj} \leq W_{jj} \leq U_{jj} \quad \forall\, j = 1, 2 \tag{19a}$$

$$L_{12}W_{12} \leq T_{12} \leq U_{12}W_{12} \tag{19b}$$

$$W_{12} \geq 0 \tag{19c}$$

$$W_{11}W_{22} = W_{12}^2 + T_{12}^2 \tag{19d}$$

where the four variables are $(W_{11}, W_{22}, W_{12}, T_{12}) \in \mathbb{R}^4$ and the data $L = (L_{11}, L_{22}, L_{12})$ and $U = (U_{11}, U_{22}, U_{12})$ satisfy $L \leq U$ and $L_{jj} \geq 0$ for $j = 1, 2$. Chen et al.'s interest in this particular convex hull arose from an analysis of the OPF problem, where (19) appears as a repeated substructure. As explained in [9], $\mathcal{J}_C$ can alternatively be expressed as the following convex hull using two complex variables $z_1, z_2 \in \mathbb{C}$:

$$\mathcal{J}_C = \overline{\text{conv}} \left\{ \begin{pmatrix} z_1 z_1^* \\ z_2 z_2^* \\ \text{Re}(z_1 z_2^*) \\ \text{Im}(z_1 z_2^*) \end{pmatrix} \in \mathbb{R}^4 : \begin{array}{c} L_{jj} \leq z_j z_j^* \leq U_{jj} \quad \forall\, j = 1, 2 \\ L_{12}\text{Re}(z_1 z_2^*) \leq \text{Im}(z_1 z_2^*) \leq U_{12}\text{Re}(z_1 z_2^*) \\ \text{Re}(z_1 z_2^*) \geq 0 \end{array} \right\}. \tag{20}$$

In particular, equation (19d) is the usual "rank-1" condition, capturing the link between the linear variables $(W_{11}, W_{22}, W_{12}, T_{12})$ and the quadratic expressions

in z_1, z_2. The authors proved that the pair of linear inequalities

$$\pi_0 + \pi_1 W_{11} + \pi_2 W_{22} + \pi_3 W_{12} + \pi_4 T_{12} \geq U_{22} W_{11} + U_{11} W_{22} - U_{11} U_{22} \quad (21a)$$
$$\pi_0 + \pi_1 W_{11} + \pi_2 W_{22} + \pi_3 W_{12} + \pi_4 T_{12} \geq L_{22} W_{11} + L_{11} W_{22} - L_{11} L_{22} \quad (21b)$$

are valid for $\mathcal{J}_C$, where

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

and where $f(x) := (\sqrt{1 + x^2} - 1)/x$ when $x > 0$ and $f(0) := 0$. In fact, they proved that (21), when added to the Shor relaxation, is sufficient to capture $\mathcal{J}_C$:

$$\mathcal{J}_C = \left\{ (W_{11}, W_{22}, W_{12}, T_{12}) : \begin{array}{c} \text{(19a)–(19c)} \\ W_{11} W_{22} \geq W_{12}^2 + T_{12}^2 \\ \text{(21)} \end{array} \right\}.$$

Here, the convex constraint $W_{11} W_{22} \geq W_{12}^2 + T_{12}^2$ is equivalent to the regular positive-semidefinite condition.

We now relate (21) to our inequalities (18). Defining

$$\mathcal{F} := \left\{ x \in \mathbb{R}^3 : \begin{array}{c} L_{11} \leq x_1^2 + x_2^2 \leq U_{11} \\ L_{22} \leq x_3^2 \leq U_{22} \\ L_{12} x_1 x_3 \leq x_2 x_3 \leq U_{12} x_1 x_3 \\ x_1 x_3 \geq 0, \quad x_3 \geq 0 \end{array} \right\}.$$

and $\mathcal{G}$ by (5), the following proposition establishes an equivalence between $\mathcal{J}_C$ and $\mathcal{G}$.

Proposition 3. $\mathcal{J}_C = \{(X_{11} + X_{22}, X_{33}, X_{13}, X_{23}) : (x, X) \in \mathcal{G}\}$.

Proof. Consider (20). Because the quadratic terms $z_1 z_1^*$, $z_2 z_2^*$, and $z_1 z_2^*$ are unaffected by a rotation of $\mathbb{C}$ applied simultaneously to both z_1 and z_2, we may enforce $\mathrm{Re}(z_2) \geq 0$ and $\mathrm{Im}(z_2) = 0$ without changing the definition of $\mathcal{J}_C$. Then writing $z_1 = x_1 + ix_2$ and $z_2 = x_3$ for $x \in \mathbb{R}^3$, we thus have $\mathcal{J}_C = \overline{\mathrm{conv}}\left\{(x_1^2 + x_2^2, x_3^2, x_1 x_3, x_2 x_3) : x \in \mathcal{F} \subseteq \mathbb{R}^3\right\}$, which proves the proposition. □

Our next proposition establishes an alternative form for $\mathcal{F}$, which matches the development in Section 2 except that the SOCs involve only two scalar variables, even though $\mathcal{F}$ is 3-dimensional. However, the results of Section 2 can easily be adapted to this case, the key point being that the Hessians of the SOCs are equal. First we need a lemma.

Lemma 1. *For $n = 2$, let $\mathcal{P} := \{x \in \mathbb{R}^2 : Ax \leq 0\}$ be a polyhedral cone with $A \in \mathbb{R}^{2\times 2}$. Then $\mathcal{P} = \{x : \|\binom{x_1}{x_2}\| \leq b^T x\}$ for some $b \in \mathbb{R}^2$.*

Proof. First assume that $\mathcal{P}$ is contained in the right side of the plane, i.e., $\mathcal{P} \subseteq \{x : x_1 \geq 0\}$ and that $\mathcal{P}$ is symmetric about the x_1 axis. Then, for some $\beta \geq 0$,

$$\begin{aligned}
\mathcal{P} &= \{x : x_1 \geq 0, -\beta x_1 \leq x_2 \leq \beta x_1\} \\
&= \{x : x_1 \geq 0, x_2^2 \leq \beta^2 x_1^2\} \\
&= \{x : x_1 \geq 0, x_1^2 + x_2^2 \leq (1+\beta^2) x_1^2\} \\
&= \{x : \|\tbinom{x_1}{x_2}\| \leq \sqrt{1+\beta^2}\, x_1\},
\end{aligned}$$

which proves the result in this case. For general $\mathcal{P}$, we may apply an orthogonal rotation to revert to the previous case, which does not affect the norm $\|\binom{x_1}{x_2}\|$ (but does change the exact form of b). □

We next state and prove the proposition. Note that the assumptions $L_{22} > 0$ and $U_{12} > L_{12}$ in the proposition are realistic for power networks: the first ensures the voltage magnitude at a bus is positive, and the second allows for a positive voltage-angle difference between the involved buses.

Proposition 4. *Suppose $L_{22} > 0$ and $U_{12} > L_{12}$. Then*

$$\mathcal{F} = \left\{ x \in \mathbb{R}^3 : \begin{array}{c} \sqrt{L_{11}} \leq \left\| \binom{x_1}{x_2} \right\| \leq \sqrt{U_{11}} \\ \left\| \binom{x_1}{x_2} \right\| \leq b_1 x_1 + b_2 x_2 \\ \sqrt{L_{22}} \leq x_3 \leq \sqrt{U_{22}} \end{array} \right\}$$

where b_1 and b_2 uniquely solve the system

$$\begin{pmatrix} 1 & L_{12} \\ 1 & U_{12} \end{pmatrix} \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} = \begin{pmatrix} \sqrt{1 + L_{12}^2} \\ \sqrt{1 + U_{12}^2} \end{pmatrix}.$$

Proof. The assumption $L_{22} > 0$ implies $x_3 > 0$, which in turn implies

$$\mathcal{F} = \left\{ x \in \mathbb{R}^3 : \begin{array}{c} L_{11} \leq x_1^2 + x_2^2 \leq U_{11} \\ \sqrt{L_{22}} \leq x_3 \leq \sqrt{U_{22}} \\ L_{12} x_1 \leq x_2 \leq U_{12} x_1 \\ x_1 \geq 0 \end{array} \right\}.$$

Next, the assumption $U_{12} > L_{12}$ makes $x_1 \geq 0$ redundant, and clearly the first constraint in $\mathcal{F}$ is equivalent to $\sqrt{L_{11}} \leq \| \binom{x_1}{x_2} \| \leq \sqrt{U_{11}}$.

To complete the proof, we claim that $L_{12} x_1 \leq x_2 \leq U_{12} x_1$ is equivalent to the SOC constraint $\| \binom{x_1}{x_2} \| \leq b_1 x_1 + b_2 x_2$. Indeed, it is clear that the set defined by these two linear inequalities is a polyhedral cone with the two extreme rays $r^1 = \binom{1}{L_{12}}$ and $r^2 = \binom{1}{U_{12}}$. So, by the lemma, the set is SOC-representable in the form $\| \binom{x_1}{x_2} \| \leq b_1 x_1 + b_2 x_2$ for some $b \in \mathbb{R}^2$. In particular, the extreme rays r^j must satisfy $\|r^j\| = b^T r^j$. By plugging in the values of r^1 and r^2, we get the 2×2 linear system defining b, as desired. Note that the 2×2 matrix is invertible because its determinant $U_{12} - L_{12}$ is positive. □

Based on Propositions 3 and 4, we now prove that (21) is simply (18) tailored to the OPF case.

Theorem 2. *Inequalities (21) are the inequalities (18) tailored to system (19).*

Proof. By Proposition 3, we can translate (21a) to the variables (x, X). After

collecting terms, (21a) becomes

$$(\pi_0+U_{11}U_{22})+(\pi_1-U_{22})(X_{11}+X_{22})+(\pi_2-U_{11})X_{33}+\pi_3X_{13}+\pi_4X_{23} \geq 0. \quad (22)$$

Using Proposition 4, consider (18a) with the following replacements:

$$x \leftarrow \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}, \;\; r \leftarrow \sqrt{L_{11}}, \;\; R \leftarrow \sqrt{U_{11}}, \;\; \lambda \leftarrow \sqrt{L_{22}}, \;\; s^T x \leftarrow x_3, \;\; \mu \leftarrow \sqrt{U_{22}}.$$

This results in the following valid inequality:

$$\left(\sqrt{L_{22}U_{22}} + U_{22}\right) \frac{X_{11} + X_{22} + \sqrt{L_{11}U_{11}}}{\sqrt{L_{11}} + \sqrt{U_{11}}} \leq \left(\sqrt{L_{22}} + \sqrt{U_{22}}\right)(b_1X_{13} + b_2X_{23}) + (U_{22} - X_{33})\sqrt{U_{11}}.$$

Simple, although tedious, algebraic manipulations establish that this inequality is precisely (22). A similar argument establishes that (21b) corresponds to (18b).[3] □

We also verified numerically that (21) is not captured by $\mathcal{R}_{\text{shor}} \cap \mathcal{R}_{\text{ksoc}}$ in this case.

3.2 Intersection of the ball and nonnegative orthant

As stated in the Introduction, the critical feature of $\mathcal{F}$ studied in this paper is its intersection of the ball with a second SOC-representable set, which shares the Hessian identity matrix. However, there are of course many other forms of $\mathcal{F}$ that can be of interest in practice. For example, when $\mathcal{F}$ is the nonnegative orthant, then $\mathcal{G}$ is the completely positive cone, which can be used to model many NP-hard problems as linear conic programs [4]. Another common case is when $\mathcal{F}$ is a box, e.g., the set $[0, 1]^n$ [6].

Let us examine the case in which $\mathcal{F}$ is the intersection of the nonnegative orthant

[3] We provide Matlab code for these manipulations at the website `https://github.com/A-Eltved/strengthened_sdr`.

and the unit ball. For general n, define $\mathcal{F} := \{x \geq 0 : \|x\| \leq 1\} \subseteq \mathbb{R}^n$. Since

$$x \in \mathcal{F} \quad \Rightarrow \quad \|x\| \leq \|x\|_1 = e^T x,$$

we have

$$\mathcal{F} \subseteq \{x : \|x\| \leq 1, \|x\| \leq e^T x\}, \tag{23}$$

and for $n = 2$, one can actually show that (23) is an equation. Since $\mathcal{F}$ is a subset of the nonnegative orthant, any inequality, which is valid for the completely positive cone, is also valid for $\mathcal{F}$, but here we focus on the implied structure in (23). Section 2 applies with $r = 0, R = 1, c = 0, b = e$, and $a = 0$. In particular, the constraints $\mathrm{tr}(X) \leq 1$ and $\mathrm{tr}(X) \leq e^T x$ are valid for $\mathcal{G}$; see the Introduction and inequality (6).

We can strengthen $\mathrm{tr}(X) \leq 1$ and $\mathrm{tr}(X) \leq e^T x$ using the slab inequalities of Section 2.1. Geometrically, given any $s \in \mathbb{R}^n$ with $s \geq 0$ and $\|s\| = 1$, we have the slab $\lambda := 0 \leq s^T x \leq 1 =: \mu$, which is valid for $\mathcal{F}$:

$$0 \leq s^T x \leq \|s\|\|x\| = \|x\| \leq 1.$$

After linearization, inequality (15) in this case reads $1 - s^T x + s^T X e \geq \mathrm{tr}(X)$. Moreover, if we switch the role of q_x and l_x in (15)—recall that q_x is linear for slabs—then we have $s^T x + e^T x - s^T X e \geq \mathrm{tr}(X)$. Rearranging, we write these two inequalities as

$$\mathrm{tr}(X) \leq 1 + s^T(Xe - x) \tag{24a}$$

$$\mathrm{tr}(X) \leq e^T x - s^T(Xe - x). \tag{24b}$$

Letting s vary over its constraints $\|s\| = 1$ and $s \geq 0$, we derive a compact SOC-representation of this class of inequalities over various domains of $\mathcal{G}$.

Theorem 3. *Let (I, J) be a partition of the index set $\{1, \ldots, n\}$, and define the domain*

$$\mathcal{D}_{IJ} := \left\{ (x, X) : \begin{array}{l} [Xe - x]_I \geq 0 \\ [Xe - x]_J \leq 0 \end{array} \right\}.$$

Then the following SOC constraints are locally valid for $\mathcal{G}$ on $\mathcal{D}_{IJ}$:

$$\operatorname{tr}(X) \leq 1 - \|[Xe - x]_J\| \tag{25a}$$

$$\operatorname{tr}(X) \leq e^T x - \|[Xe - x]_I\|. \tag{25b}$$

Moreover, (25) imply all valid inequalities (15) derived from slabs of the form $0 \leq s^T x \leq 1$, where s is any vector satisfying $\|s\| = 1$ and $s \geq 0$.

Proof. Consider the constraints (24), and for notational convenience, define $y := Xe - x$. Because $s \geq 0$, the quantity $s^T y$ on the right-hand side of (24a) breaks into $s_I^T y_I \geq 0$ and $s_J^T y_J \leq 0$ on $\mathcal{D}_{IJ}$. By minimizing the right-hand side of (24a) with respect to s, we achieve the tightest cut corresponding to $s = (s_I, s_J) = (0, -y_J/\|y_J\|)$, which yields $\operatorname{tr}(X) \leq 1 - \|y_J\|$, as desired. A similar argument for (24b) yields $\operatorname{tr}(X) \leq e^T x - \|y_I\|$. □

We remark that, when I is empty, inequality (25b) reduces to the inequality $\operatorname{tr}(X) \leq e^T x$ over D_{IJ}. Similarly, when J is empty, (25a) is $\operatorname{tr}(X) \leq 1$.

In practice, one idea for using Theorem 3 is as follows. For a given relaxation in (x, X), solve the relaxation to obtain an optimal solution $(\bar{x}, \bar{X})$. Then define the partition (I, J) and corresponding domain D_{IJ} according to $\bar{X}e - \bar{x}$. Then, if either of the inequalities in (25) is violated, we can derive a violated supporting hyperplane of the SOC constraint. After adding the violated linear inequality to the current relaxation, which is globally valid because it is linear, we can resolve and repeat the process.

We close this section with an example showing that the cuts derived above are not implied by $\mathcal{R}_{\text{shor}} \cap \mathcal{R}_{\text{ksoc}}$.

Example 2. *Let $n = 2$, and consider $I = \{1, 2\}$ and $J = \emptyset$. Then $\operatorname{tr}(X) \leq e^T x - \|Xe - x\|$ is valid on the domain $\mathcal{D}_{IJ} = \{(x, X) : Xe - x \geq 0\}$. In particular, $\operatorname{tr}(X) \leq e^T x - u^T(Xe - x)$ for all vectors u satisfying $\|u\| = 1$, and taking $u = e_1$, we have $\operatorname{tr}(X) \leq e^T x - [Xe - x]_1$, which is globally valid since it is linear. Minimizing $e^T x - [Xe - x]_1 - \operatorname{tr}(X)$ over $\mathcal{R}_{shor} \cap \mathcal{R}_{ksoc}$ yields the optimal value -0.088562, indicating that $\mathcal{R}_{shor} \cap \mathcal{R}_{ksoc}$ does not capture this valid constraint.*

4 Separation

In this section, we argue that the inequalities (12)–(13) given by Theorem 1 and Corollary 1 are separable in polynomial time. To state this result precisely, we assume that $[c]_{\max}$ has already been pre-computed and that a fixed convex relaxation of the convex hull $\mathcal{G}$ defined by (5) is available. For convenience, we write this fixed convex relaxation

$$\mathcal{R} := \left\{(x, X) : Y(x, X) \in \widehat{\mathcal{R}}\right\} \supseteq \mathcal{G},$$

where $Y(x, X)$ is given by (3) and $\widehat{\mathcal{R}}$ is a closed, convex cone in the space of $(n + 1) \times (n + 1)$ symmetric matrices. In particular, $\mathcal{R}$ is just the slice of $\widehat{\mathcal{R}}$ with the top-left corner of Y set to 1. Then the relaxation of (1) over $\mathcal{R}$ can be stated as $\min\{H \bullet X + 2\, g^T x : (x, X) \in \mathcal{R}\}$ with dual

$$\max\left\{y : \begin{pmatrix} -y & g^T \\ g & H \end{pmatrix} \in \widehat{\mathcal{R}}^*\right\}$$

where $\widehat{\mathcal{R}}^*$ is the dual cone of $\widehat{\mathcal{R}}$. We state this general form for ease of notation and to make evident that one can choose different $\mathcal{R}$ in computation. For example, one could take $\mathcal{R} = \mathcal{R}_{\text{shor}}$ at one extreme or $\mathcal{R} = \mathcal{R}_{\text{shor}} \cap \mathcal{R}_{\text{ksoc}}$ at the other.

In fact, to separate (12)–(13) we will use the following observation concerning $\mathcal{R}$, $\widehat{\mathcal{R}}$, and $\widehat{\mathcal{R}}^*$:

Observation. *Given a quadratic function $q(x) := x^T H_q x + 2\, g_q^T x + f_q$, if there exists $y \in \mathbb{R}$ such that*

$$\begin{pmatrix} -y + f_q & g_q^T \\ g_q & H_q \end{pmatrix} \in \widehat{\mathcal{R}}^*,$$

then $q(x) \geq y$ for all $x \in \mathcal{F}$.

This observation follows by weak duality because y is a lower bound on the optimal relaxation value of $H_q \bullet X + 2\, g_q^T x + f_q$ over $(x, X) \in \mathcal{R}$, which is itself a lower bound on the minimum value of $q(x)$ over $x \in \mathcal{F}$. As a result, the following system guarantees that the conditions of Theorem 1 on $q(x)$ and $l(x)$

hold, where (H_q, g_q, f_q), (g_l, f_l), and $[q+l]_{\min}$ are the variables:

$$\begin{pmatrix} f_q & g_q^T \\ g_q & H_q \end{pmatrix} \in \widehat{\mathcal{R}}^*, \quad \begin{pmatrix} f_l & g_l^T \\ g_l & 0 \end{pmatrix} \in \widehat{\mathcal{R}}^*, \tag{26a}$$

$$[q+l]_{\min} \geq 0, \quad \begin{pmatrix} -[q+l]_{\min} + f_q + f_l & (g_q + g_l)^T \\ g_q + g_l & H_q \end{pmatrix} \in \widehat{\mathcal{R}}^*. \tag{26b}$$

Then, separation amounts to optimizing the linear function in (12)—or (13) as the case may be—over (26) for fixed values of (x, X). However, before we state the exact separation problem for (12), we require one additional assumption, namely that $\mathcal{F}$ is full-dimensional, i.e., there exists $\hat{x} \in \mathcal{F}$ such that $\|\hat{x}\| < 1$ and $\|\hat{x} - c\| < b^T x - a$. In this case, it is well known that $\mathcal{G}$ and hence $\mathcal{R}$ are also full-dimensional in (x, X)-space. In particular, $(\hat{x}, \hat{x}\hat{x}^T) \in \operatorname{int}(\mathcal{G}) \subseteq \operatorname{int}(\mathcal{R})$, and hence

$$\hat{Y} := \begin{pmatrix} 1 \\ \hat{x} \end{pmatrix} \begin{pmatrix} 1 \\ \hat{x} \end{pmatrix}^T \in \operatorname{int}(\widehat{\mathcal{R}}).$$

It thus follows by standard duality theory that $\widehat{\mathcal{R}}^* \cap \{J : \hat{Y} \bullet J \leq 1\}$ is a bounded truncation of $\widehat{\mathcal{R}}^*$. This truncation is important so that the separation problem below is bounded and thus has a well-defined optimal value.

We are now ready to state the separation subproblem for (12) given fixed values $(\bar{x}, \bar{X})$ of the variables (x, X):

$$\begin{aligned} \min \quad & (r+R)R\left(H_q \bullet \bar{X} + 2\, g_q^T \bar{x} + f_q\right) \\ & + (r+R)\left(2\, g_l b^T \bullet \bar{X} + (f_l b - 2a g_l)^T \bar{x} - a f_l\right) \\ & - [q+l]_{\min} \operatorname{tr}(\bar{X}) - rR\left(H_q \bullet \bar{X} + 2(g_q + g_l)^T \bar{x} + (f_q + f_l)\right) \\ & + \left(2\, g_l c^T \bullet \bar{X} + f_l c^T \bar{x}\right) + [c]_{\max} R(2\, g_l^T \bar{x} + f_l) \end{aligned} \tag{27a}$$

$$\text{s.t.} \quad (26) \tag{27b}$$

$$\hat{Y} \bullet \begin{pmatrix} f_q & g_q^T \\ g_q & H_q \end{pmatrix} \leq 1, \quad \hat{Y} \bullet \begin{pmatrix} f_l & g_l^T \\ g_l & 0 \end{pmatrix} \leq 1. \tag{27c}$$

The subproblem for (13) is similar—just replace r with 0.

We remark that system (26) could be simplified in certain cases. For example,

if $r = 0$ and hence $\mathcal{F}$ is convex, then it is not difficult to see that the second condition of (26a), which ensures that $l(x)$ is nonnegative over $\mathcal{F}$, could be replaced by a dual system based on $\mathcal{F}$ alone, not on $\mathcal{R}$. One could also simplify by forcing additional structure on $q(x)$ and $l(x)$. For example, one could separate against the slabs $\lambda \leq s^T x \leq \mu$ introduced in Section 2.1 by forcing $(H_q, g_q, f_q) = (0, -\frac{1}{2}s, \mu)$, $(g_l, f_l) = (\frac{1}{2}s, -\lambda)$, and $[q + l]_{\min} = \mu - \lambda$, in which case (26b) is automatically satisfied.

The following example demonstrates the separation procedure, whose implementation will be discussed in the next section:

Example 3. *Consider the* 2*-dimensional problem*

$$\begin{aligned} \min \quad & -x_1^2 - x_2^2 - 1.1x_1 - x_2 \\ s.t. \quad & \|x\| \leq 1 \\ & \|x\| \leq 1 - x_1 - x_2 \end{aligned}$$

with $H = -I$, $g = (-0.55, -0.5)$, $r = 0$, $R = 1$, $a = -1$, $b = (-1, -1)$, *and* $c = (0, 0)$ *in* (1). *All values reported here are truncated from the computations and therefore approximate. The optimal value of* $\min\{H \bullet X + 2g^T x : (x, X) \in \mathcal{R}_{shor} \cap \mathcal{R}_{ksoc}\}$ *is* -1.1431 *with optimal solution*

$$\bar{x} = \begin{pmatrix} 0.2922 \\ -0.1783 \end{pmatrix}, \ \bar{X} = \begin{pmatrix} 0.4963 & -0.3210 \\ -0.3210 & 0.5037 \end{pmatrix}.$$

Solving the separation subproblem at $(\bar{x}, \bar{X})$, *we obtain the cut corresponding to*

$$\begin{aligned} q_1(x) &= x^T \begin{pmatrix} -0.3812 & 0 \\ 0 & -0.3812 \end{pmatrix} x + 2 \begin{pmatrix} -0.5578 \\ -0.5531 \end{pmatrix} x + 0.8563, \\ l_1(x) &= 2 \begin{pmatrix} 0.3462 \\ 0.3608 \end{pmatrix} x + 1, \\ [q_1 + l_1]_{\min} &= 1.42. \end{aligned}$$

We add the corresponding cut, resolve to obtain a new $(\bar{x}, \bar{X})$, *and repeat this*

loop two more times, resulting in the cuts

$$
\begin{aligned}
q_2(x) &= x^T \begin{pmatrix} -0.7065 & 0.1719 \\ 0.1719 & -0.4368 \end{pmatrix} x + 2 \begin{pmatrix} -0.7808 \\ -0.7278 \end{pmatrix} x + 1, \\
l_2(x) &= 2 \begin{pmatrix} 0.3442 \\ 0.3626 \end{pmatrix} x + 1, \\
[q_2 + l_2]_{\min} &= 1.155, \\
q_3(x) &= x^T \begin{pmatrix} -0.6296 & 0.2398 \\ 0.2398 & -0.4512 \end{pmatrix} x + 2 \begin{pmatrix} -0.7868 \\ -0.7580 \end{pmatrix} x + 1, \\
l_3(x) &= 2 \begin{pmatrix} 0.3479 \\ 0.3591 \end{pmatrix} x + 1, \\
[q_3 + l_3]_{\min} &= 1.149.
\end{aligned}
$$

We finally obtain the rank-1, and hence optimal, solution

$$
Y(x^\star, X^\star) = \begin{pmatrix} 1 & 0.7071 & -0.7071 \\ 0.7071 & 0.5 & -0.5 \\ -0.7071 & -0.5 & 0.5 \end{pmatrix}
$$

with objective value -1.0707. *We note that, even though the procedure generates three cuts, the last cut is actually enough to recover the rank-1 solution. Moreover, running this procedure starting from* $\mathcal{R}_{shor}$ *instead of* $\mathcal{R}_{shor} \cap \mathcal{R}_{ksoc}$, *we also get the same optimal* $(x^\star, X^\star)$ *after adding 16 cuts.*

5 Computational Results

To quantify the practical effect of the cuts proposed in Theorem 1 and Corollary 1, we embed the separation subproblem described in Section 4 in a straightforward implementation to solve random instances of the form (1). We consider two relaxations to "bootstrap" the separation procedure: $\mathcal{R}_{\text{shor}}$ and $\mathcal{R}_{\text{shor}} \cap \mathcal{R}_{\text{ksoc}}$. We will denote by $\mathcal{R}_{\text{cuts}}$ the points (x, X) satisfying the added cuts, so that our improved relaxations will be expressed as $\mathcal{R}_{\text{shor}} \cap \mathcal{R}_{\text{cuts}}$ and $\mathcal{R}_{\text{shor}} \cap \mathcal{R}_{\text{ksoc}} \cap \mathcal{R}_{\text{cuts}}$.

We implement our experiments in Matlab 9.6 (R2019a) using CVX [11] to model the relaxations and MOSEK 9.1 [17] to solve them. We run the problem instances on a single core of an Intel Xeon E5-2650v4 processor using a maximum of 2GB memory. We do not report complete run times because we are most interested in the strength of the added cuts, but we do report the number of cuts added to measure the overall effort. Recall that calculating a single cut requires solving the separation problem (27) described in Section 4, which in essence involves three copies of the current bootstrap relaxation—$\mathcal{R}_{\text{shor}} \cap \mathcal{R}_{\text{cuts}}$ or $\mathcal{R}_{\text{shor}} \cap \mathcal{R}_{\text{ksoc}} \cap \mathcal{R}_{\text{cuts}}$. However, to give the reader a sense of the run times, consider the following: for an instance of our largest dimension, $n = 10$, solving $\mathcal{R}_{\text{shor}}$ took approximately 0.6 seconds, solving $\mathcal{R}_{\text{shor}} \cap \mathcal{R}_{\text{ksoc}}$ required about 50 seconds, and solving a single separation problem for $\mathcal{R}_{\text{shor}} \cap \mathcal{R}_{\text{ksoc}}$ took approximately 64 seconds. We note that our implementation is rudimentary and makes no effort to take advantage of, for example, any particular problem structure or sparsity, so these times can probably be improved significantly.

We generate a single random instance by fixing the dimension n and generating random data a, b, c, r, R, H, g in such a way that (1) is feasible with a known interior point $\hat{x}$, which is also randomly generated. In short, we first set $R = 1$ without loss of generality, generate r uniformly in $[0, R]$, generate $\hat{x}$ uniformly in $\{x : r \leq \hat{x} \leq R\}$, generate b, c, H, g with entries i.i.d. standard normal, and finally set $a := b^T\hat{x} - \|\hat{x} - c\| - \theta$, where θ is uniform in $[0, 1]$ so that $\mathcal{F}$ has a nonempty interior.[4] Recall that $\hat{x}$ is required for the separation procedure as discussed in Section 4. Before running the separation procedure for an instance, we compute $[c]_{\max}$ by a binary search on $[c]_{\max}$ over the interval $[0, \|c\|]$ as discussed in Section 2. Then, when running the overall algorithm, we consider the current relaxation's optimal solution $(\bar{x}, \bar{X})$ to be *separated* if: the objective value of the separation subproblem (27) is less than $\tau_{\text{sep}} = -10^{-5}$; or the optimal value of the separation subproblem for the inequalities (13) in Corollary 1, i.e., (27) with $r = 0$, is less than τ_{sep}. If $(\bar{x}, \bar{X})$ is indeed separated, we add the resulting cut represented by the data $(H_q, g_q, f_q, g_l, f_l, [q + l]_{\min})$ to the current bootstrap relaxation, optimize for a new point to be separated, and repeat. The overall loop stops when the current $(\bar{x}, \bar{X})$ is not separated with tolerance τ_{sep}.

[4] We refer the reader to our GitHub site (`https://github.com/A-Eltved/strengthened_sdr`) for the full random-generation procedure.

Regarding a given relaxation and its optimal solution $(\bar{x}, \bar{X})$, we say the relaxation is *exact* if $Y(\bar{x}, \bar{X})$ satisfies

$$\frac{\lambda_1(Y(\bar{x}, \bar{X}))}{\lambda_2(Y(\bar{x}, \bar{X}))} > \tau_{\text{rank}},$$

where $\lambda_1(M)$ denotes the largest eigenvalue of M, $\lambda_2(M)$ denotes the second largest eigenvalue of M, and $\tau_{\text{rank}} > 0$ is a tolerance, which we choose to be 10^4 in our implementation, ensuring that $Y(\bar{x}, \bar{X})$ is numerically rank-1. We define the *gap* as the difference between the optimal value of (1) and the relaxation optimal value. Note that an exact relaxation implies a gap of 0.

After running the algorithm on a particular instance, we classify the instance into one of two categories: *exact initial* or *inexact initial*, when the initial bootstrap relaxation is exact or inexact, respectively. Furthermore, we break all inexact-initial instances into one of three subcategories: *improved*, when the initial relaxation gap is improved but not completely closed to 0; *closed*, when the relaxation becomes exact after adding one or more cuts; and *no improvement*, when no cuts are successfully added to improve the gap, i.e., the separation routine does not help. (Actually, in the tables below, we will not directly report information about the exact-initial and no-improvement instances, as these details will be implicitly available from the other categories.)

We conduct these experiments for several values of n and many randomly generated instances. In addition, we also consider special cases where some of the data a, b, c, r, R is fixed to zero in order to assess whether the cuts are more effective in these special cases. In particular, we consider the following three cases: the general case, where no data is fixed *a priori* to zero; the special case with $r = a = 0$ and $c = 0$; and the case of the TTRS (two trust region subproblem) with $r = 0$ and $b = 0$. For each of these cases, we generate 15,000 instances for each dimension $2 \leq n \leq 10$, and we solve each instance twice, once bootstrapping from $\mathcal{R}_{\text{shor}}$ and once from $\mathcal{R}_{\text{shor}} \cap \mathcal{R}_{\text{ksoc}}$.

For the improved and closed instances, we report the average number of cuts added. Also for the improved instances, we report the average gap closure in percentage terms, i.e., we report the average relative gap closure. Since we

do not actually know the optimal value of (1) for the improved instances, to approximate the relative gap closure from above, we calculate a local minimum value, v_{local}, by taking the lowest value of the quadratic objective function gotten by running Matlab's `fmincon` with 100 random initial points. The relative gap for the instance is then calculated as

$$\text{relative gap closure} = \frac{v_{\text{relax final}} - v_{\text{relax initial}}}{v_{\text{local}} - v_{\text{relax initial}}} \times 100\%,$$

where $v_{\text{relax initial}}$ is the optimal value of the initial relaxation and $v_{\text{relax final}}$ is the optimal value of the final relaxation.

5.1 The general case

We consider 15,000 random instances for each dimension $2 \leq n \leq 10$ and report the results separately for the $\mathcal{R}_{\text{shor}}$ and $\mathcal{R}_{\text{shor}} \cap \mathcal{R}_{\text{ksoc}}$ bootstrap relaxations in Tables 1 and 2, respectively.

In Table 1, we see that our cuts improve the $\mathcal{R}_{\text{shor}}$ relaxation in many instances. For $n = 2$, it improves more than a third of the inexact instances, and it closes the gap for about 9%. As the dimension goes up, these proportions go down, suggesting that our cuts are more effective in lower dimensions.

n	Inexact initial	Improved	Avg cuts	Avg gap closure	Closed	Avg cuts
2	2923	1188	15	51%	264	4
3	2582	761	17	46%	175	7
4	2161	422	10	40%	53	7
5	1801	416	10	36%	46	9
6	1583	265	12	36%	29	8
7	1360	186	11	36%	10	11
8	1091	140	14	39%	15	7
9	1029	107	12	34%	4	15
10	896	86	13	30%	4	11

Table 1: Results for the $\mathcal{R}_{\text{shor}}$ bootstrap relaxation on 15,000 random general instances for each dimension n. The columns *Inexact initial*, *Improved*, and *Closed* report the number of instances out of 15,000 in each category.

Table 2 shows that $\mathcal{R}_{\text{shor}} \cap \mathcal{R}_{\text{ksoc}}$ is generally quite strong for instances of the form (1). Especially for larger n, the number of inexact instances is small, and the ability of our cuts to improve or close the gaps is limited. In particular, for $n \geq 4$ our cuts do not improve any of the inexact instances, which again suggests that the cuts are most helpful in lower dimensions.

n	Inexact initial	Improved	Avg cuts	Avg gap closure	Closed	Avg cuts
2	251	40	13	45%	3	3
3	84	5	36	48%	0	—
4	44	0	—	—	0	—
5	16	0	—	—	0	—
6	6	0	—	—	0	—
7	7	0	—	—	0	—
8	2	0	—	—	0	—
9	3	0	—	—	0	—
10	3	0	—	—	0	—

Table 2: Results for the $\mathcal{R}_{\text{shor}} \cap \mathcal{R}_{\text{ksoc}}$ bootstrap relaxation on the same 15,000 random general instances as depicted in Table 1 for each dimension n. The columns *Inexact initial*, *Improved*, and *Closed* report the number of instances out of 15,000 in each category.

5.2 Special case: $r = a = 0$ and $c = 0$

We next consider the special case when $\mathcal{F}$ equals $\{x \in \mathbb{R}^n : \|x\| \leq 1, \|x\| \leq b^T x\}$ with $b \in \mathbb{R}^n$. Note that, by rotating the feasible space, we may assume without loss of generality that b lies in the direction of e, the all ones vector. In particular, we generate instances with $b = \beta e$, where $\beta \in [1/\sqrt{n}, 1/\sqrt{n} + 2n]$. The choice of this interval for β is based on the following observation: for $\beta < 1/\sqrt{n}$ the feasible space $\mathcal{F}$ is empty; for $\beta = 1/\sqrt{n}$ the feasible space $\mathcal{F}$ has no interior; for $\beta \to \infty$, the constraint $\|x\| \leq b^T x$ resembles the half space $0 \leq e^T x$.

Similar to Tables 1–2 of the previous subsection, Tables 3–4 contain the results of our separation algorithm on 15,000 randomly generated instances for each dimension, where Table 3 corresponds to $\mathcal{R}_{\text{shor}}$ and Table 4 to $\mathcal{R}_{\text{shor}} \cap \mathcal{R}_{\text{ksoc}}$. Contrary to what we saw in the general case in Tables 1–2, there does *not* seem to be a drop in the proportion of instances where the cuts help as n increases.

Overall, our cuts seem to be quite effective in this special case.

n	Inexact initial	Improved	Avg cuts	Avg gap closure	Closed	Avg cuts
2	7744	2755	22	82%	4988	2
3	7635	914	23	86%	6495	3
4	7736	395	13	83%	6966	3
5	7709	401	4	81%	6596	3
6	7584	402	5	67%	7182	3
7	7648	185	5	87%	7463	3
8	7614	131	8	89%	7483	3
9	7566	77	7	93%	7489	2
10	7552	44	7	89%	7508	2

Table 3: Results for the $\mathcal{R}_{\text{shor}}$ bootstrap relaxation on 15,000 random instances with $r = a = 0$ and $c = 0$ for each dimension n. The columns *Inexact initial*, *Improved*, and *Closed* report the number of instances out of 15,000 in each category.

n	Inexact initial	Improved	Avg cuts	Avg gap closure	Closed	Avg cuts
2	15	0	—	—	15	2
3	50	7	43	37%	30	2
4	36	4	78	75%	28	2
5	29	0	—	—	27	3
6	15	3	8	88%	12	3
7	13	2	4	57%	11	2
8	12	0	—	—	12	2
9	6	0	—	—	5	1
10	6	0	—	—	5	3

Table 4: Results for the $\mathcal{R}_{\text{shor}} \cap \mathcal{R}_{\text{ksoc}}$ bootstrap relaxation on the same 15,000 random instances as depicted in Table 3 with $r = a = 0$ and $c = 0$ for each dimension n. The columns *Inexact initial*, *Improved*, and *Closed* report the number of instances out of 15,000 in each category.

Specifically for $n = 2$, the results in Table 4 suggest that $\mathcal{R}_{\text{shor}} \cap \mathcal{R}_{\text{ksoc}} \cap \mathcal{R}_{\text{cuts}}$ is tight, i.e., it captures the convex hull $\mathcal{G}$. To test this further, we generated an additional 110,000 instances with $n = 2$. The $\mathcal{R}_{\text{shor}} \cap \mathcal{R}_{\text{ksoc}}$ relaxation was exact for 109,938 of these, and our cuts closed the gap for the remaining 62 instances with an average of 3 cuts added. Our computational experience thus motivates a conjecture:

Conjecture 1. *For the 2-dimensional feasible space* $\mathcal{F} := \{x \in \mathbb{R}^2 : \|x\| \leq$

$1, \|x\| \leq b^T x\}$ *with arbitrary* $b \in \mathbb{R}^2$, $\mathcal{R}_{shor} \cap \mathcal{R}_{ksoc} \cap \mathcal{R}_{cuts}$ *equals the convex hull* $\mathcal{G}$ *defined in* (5).

In addition, in Section 3.2, for $n = 2$ and $b = e$, we proposed the locally valid cuts (25), which were derived from slabs of a particular form. (Note that these cuts would not necessarily be valid for a different scaling $b = \beta e$.) By generating many random objectives, we were able to find 100 additional instances, which were *not* solved exactly by $\mathcal{R}_{\text{shor}} \cap \mathcal{R}_{\text{ksoc}}$, and then separated just these locally valid cuts—instead of the more general cuts represented by $\mathcal{R}_{\text{cuts}}$. All 100 instances were solved exactly, i.e., achieved the tolerance τ_{rank}. We believe this is strong evidence to support the following conjecture as well:

Conjecture 2. *For the 2-dimensional feasible space* $\mathcal{F} := \{x \in \mathbb{R}^2 : \|x\| \leq 1, \|x\| \leq e^T x\} = \{x \geq 0 : \|x\| \leq 1\}$, *the constraints defined by* $\mathcal{R}_{shor} \cap \mathcal{R}_{ksoc}$ *intersected with the locally valid cuts (25) capture the convex hull* $\mathcal{G}$ *defined in* (5).

5.3 Special case: TTRS ($b = 0$ and $r = 0$)

Setting $b = 0$ and $r = 0$ in (1) with $a < 0$ to ensure feasibility, we explore the two-trust-region subproblem (TTRS). We generate 15,000 random instances of this type for each dimension $2 \leq n \leq 10$ and bootstrap from the $\mathcal{R}_{\text{shor}}$ and $\mathcal{R}_{\text{shor}} \cap \mathcal{R}_{\text{ksoc}}$ relaxations. The results are shown in Tables 5 and 6. The trends in these tables are similar to what we saw in the general case in Section 5.1. In particular, our cuts are less effective in higher dimensions.

We catalog the following example showing an explicit case for $n = 2$ in which our cuts close the gap for TTRS compared to just applying $\mathcal{R}_{\text{shor}} \cap \mathcal{R}_{\text{ksoc}}$.

Example 4. *Consider the instance with* $n = 2$, $r = 0$, $R = 1$, $a = -0.77$, *and*

$$b = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \quad c = \begin{pmatrix} -0.38 \\ 0.18 \end{pmatrix}, \quad H = \begin{pmatrix} -1.32 & 0.21 \\ 0.21 & -0.81 \end{pmatrix}, \quad g = \begin{pmatrix} -0.25 \\ 0.05 \end{pmatrix}.$$

The (approximate) optimal value of $\min\{H \bullet X + 2\, g^T x : (x, X) \in \mathcal{R}_{shor} \cap \mathcal{R}_{ksoc}\}$ *is* -0.9087 *and the solution is not rank-1. Solving the separation problem*

n	Inexact initial	Improved	Avg cuts	Avg gap closure	Closed	Avg cuts
2	1404	364	16	33%	86	4
3	1287	172	15	27%	34	4
4	985	79	12	27%	20	5
5	745	34	9	22%	7	3
6	508	14	7	22%	3	2
7	454	4	5	25%	2	3
8	347	5	8	58%	0	—
9	293	0	—	—	1	2
10	251	1	4	2%	0	—

Table 5: Results for the $\mathcal{R}_{\text{shor}}$ bootstrap relaxation on 15,000 random TTRS instances for each dimension n. The columns *Inexact initial*, *Improved*, and *Closed* report the number of instances out of 15,000 in each category.

starting from this relaxation, we obtain the (approximate) cut corresponding to

$$g_l = \begin{pmatrix} 1.8633 \\ -0.8826 \end{pmatrix}, \quad f_l = 4.1236, \quad [q + l]_{\min} = 1.2604,$$

$$H_q = \begin{pmatrix} -4.9035 & 0.0000 \\ 0.0000 & -4.9035 \end{pmatrix}, \quad g_q = \begin{pmatrix} -1.8633 \\ 0.8826 \end{pmatrix}, \quad f_q = 2.0403.$$

Solving the relaxation with this cut, results in the (numerically) rank-1 solution

$$Y(x^\star, X^\star) = \begin{pmatrix} 1.0000 & -0.9065 & 0.4223 \\ -0.9065 & 0.8217 & -0.3828 \\ 0.4223 & -0.3828 & 0.1783 \end{pmatrix}$$

with (approximate) optimal value -0.8943.

6 Conclusions

In this paper, we have derived a new class of valid linear inequalities for SDP relaxations of problem (1). These cuts are separable in polynomial time, which, by the equivalence of separation and optimization, ensures that the SDP relaxation enforcing all of these inequalities is polynomial-time solvable. We have

n	Inexact initial	Improved	Avg cuts	Avg gap closure	Closed	Avg cuts
2	31	4	20	24%	0	—
3	78	7	43	29%	1	7
4	63	3	55	19%	0	—
5	34	1	59	6%	0	—
6	22	0	—	—	0	—
7	16	0	—	—	0	—
8	14	0	—	—	0	—
9	6	0	—	—	0	—
10	4	0	—	—	0	—

Table 6: Results for the $\mathcal{R}_{\text{shor}} \cap \mathcal{R}_{\text{ksoc}}$ bootstrap relaxation on the same 15,000 random TTRS instances as depicted in Table 5 for each dimension n. The columns *Inexact initial*, *Improved*, and *Closed* report the number of instances out of 15,000 in each category.

also shown that a special case of our cuts has been applied by Chen et al. [9] to obtain the convex hull of an important substructure arising in the OPF problem. In addition, we have extended our methodology to derive new, locally valid, second-order-cone cuts for nonconvex quadratric programs over the mixed polyhedral-conic set $\{x \geq 0 : \|x\| \leq 1\}$. Using specific examples as well as computational experiments, we have demonstrated that the new class of valid inequalities strengthens the strongest known SDP relaxation, $\mathcal{R}_{\text{shor}} \cap \mathcal{R}_{\text{ksoc}}$, especially in low dimensions.

For the specific 2-dimensional feasible set $\mathcal{F} = \{x \in \mathbb{R}^2 : \|x\| \leq 1, x \leq b^T x\}$, our computational experiments indicate that our cuts intersected with $\mathcal{R}_{\text{shor}} \cap \mathcal{R}_{\text{ksoc}}$ capture the relevant convex hull $\mathcal{G}$. We leave this as a conjecture requiring further research. Furthermore, when $b = e$, we also conjecture that the locally valid cuts (25), which are derived from slabs, are by themself enough to capture $\mathcal{G}$. For general $\mathcal{F}$, however, our cuts do not close the gap fully, and so there remains room for improvement.

One limitation of our approach is the assumption that the SOC constraint (1c) shares the identity Hessian with the hollow ball (1b). If instead we are presented with a general SOC constraint $\|Jx - c\| \leq b^T x - a$, where $J \in \mathbb{R}^{n \times n}$ is arbitrary,

one idea would be to bound

$$\begin{aligned} b^T x - a &\geq \|Jx - c\| \\ &\geq \|x - c\| - \|x - Jx\| \\ &= \|x - c\| - \|(I - J)x\| \\ &\geq \|x - c\| - \sqrt{\lambda_{\max}[(I - J)^T(I - J)]}\, R, \end{aligned}$$

which yields the valid constraint $\|x-c\| \leq b^T x - \left(a - \sqrt{\lambda_{\max}[(I - J)^T(I - J)]}R\right)$, to which our methodology can be applied. Additional options for handling arbitrary Hessians can be considered by refining the deriviations of Section 2.

Further opportunities for future research include streamlining the separation subroutine, investigating the effectiveness of our cuts in higher dimensions, and examining other applications where the structure of (1) appears. Also, the idea of using the self-duality of a cone to derive valid linear cuts could be applied to other self-dual cones or possibly even non-self-dual cones.

Acknowledgments

The authors acknowledge the support of their respective universities, which allowed the first author to visit the second author in 2019–20, when this research was initiated.

References

[1] K. M. Anstreicher. Kronecker product constraints with an application to the two-trust-region subproblem. *SIAM Journal on Optimization*, 27(1):368–378, 2017.

[2] C. J. Argue, F. Kılınç-Karzan, and A. L. Wang. Necessary and sufficient conditions for rank-one generated cones. Technical report, 2020.

[3] D. Bienstock. A note on polynomial solvability of the CDT problem. *SIAM Journal on Optimization*, 26(1):488–498, 2016.

[4] S. Burer. On the copositive representation of binary and continuous nonconvex quadratic programs. *Math. Program.*, 120(2, Ser. A):479–495, 2009.

[5] S. Burer and K. M. Anstreicher. Second-order-cone constraints for extended trust-region subproblems. *SIAM Journal on Optimization*, 23(1):432–451, 2013.

[6] S. Burer and A. N. Letchford. On nonconvex quadratic programming with box constraints. *SIAM Journal on Optimization*, 20(2):1073–1089, 2009.

[7] S. Burer and B. Yang. The trust region subproblem with non-intersecting linear constraints. *Mathematical Programming*, 149(1-2, Ser. A):253–264, 2015.

[8] M. R. Celis, J. E. Dennis, and R. A. Tapia. A trust region strategy for nonlinear equality constrained optimization. In *Numerical optimization, 1984 (Boulder, Colo., 1984)*, pages 71–82. SIAM, Philadelphia, PA, 1985.

[9] C. Chen, A. Atamtürk, and S. S. Oren. A spatial branch-and-cut method for nonconvex QCQP with bounded complex variables. *Mathematical Programming*, 165(2, Ser. A):549–577, 2017.

[10] A. R. Conn, N. I. M. Gould, and P. L. Toint. *Trust-Region Methods*. MPS/SIAM Series on Optimization. SIAM, Philadelphia, PA, 2000.

[11] CVX Research, Inc. CVX: Matlab software for disciplined convex programming, version 2.1. `http://cvxr.com/cvx`, Dec. 2018.

[12] A. Eltved. *Convex Relaxation Techniques for Nonlinear Optimization*. PhD thesis, Technical University of Denmark, 2021.

[13] R. Jiang and D. Li. Second order cone constrained convex relaxations for nonconvex quadratically constrained quadratic programming. *Journal of Global Optimization*, 75(2):461–494, June 2019.

[14] J. B. Lasserre. Global optimization with polynomials and the problem of moments. *SIAM J. Optim.*, 11(3):796–817, 2000/01.

[15] S. H. Low. Convex relaxation of optimal power flow—Part I: Formulations and Equivalence. *IEEE Transactions on Control of Network Systems*, 1(1):15–27, 2014.

[16] S. H. Low. Convex relaxation of optimal power flow—Part II: Exactness. *IEEE Transactions on Control of Network Systems*, 1(2):177–189, 2014.

[17] MOSEK ApS. *The MOSEK optimization toolbox for MATLAB manual. Version 9.0.105*, 2019.

[18] H. D. Sherali and W. P. Adams. *A reformulation-linearization technique for solving discrete and continuous nonconvex problems*, volume 31 of *Nonconvex Optimization and its Applications*. Kluwer Academic Publishers, Dordrecht, 1999.

[19] N. Z. Shor. Quadratic optimization problems. *Soviet Journal of Computer and Systems Sciences*, 25:1–11, 1987.

[20] R. J. Stern and H. Wolkowicz. Indefinite trust region subproblems and nonsymmetric eigenvalue perturbations. *SIAM Journal on Optimization*, 5(2):286–313, 1995.

[21] Y. Ye and S. Zhang. New results on quadratic minimization. *SIAM Journal on Optimization*, 14(1):245–267, 2003.

Appendix C

Paper C

---

[31] Anders Eltved and Martin S. Andersen. "Sufficient Conditions for Exact Semidefinite Relaxation of Homogeneous Quadratically Constrained Quadratic Programs with Forest Structure". Submitted. 2020

Status: Submitted.

Sufficient Conditions for Exact Semidefinite Relaxation of Homogeneous Quadratically Constrained Quadratic Programs with Forest Structure

Anders Eltved*† Martin S. Andersen†

December 18, 2020

Abstract

We study the semidefinite programming relaxation of nonconvex quadratically constrained quadratic programs without linear terms and whose aggregate sparsity graph is a forest. We present sufficient conditions for the semidefinite relaxation to be exact which implies that the original problem is solvable in polynomial time. The conditions comprise a family of second-order cone programs that involve some (but not all) problem data and can be solved in polynomial time. As an extension, we propose a robust exactness guarantee that applies to a family of similar problems defined by a region around the nominal data where the relaxation remains exact.

*Corresponding author. E-mail: anderseltved@gmail.com

†Department of Applied Mathematics and Computer Science, Technical University of Denmark, 2800 Kgs. Lyngby, Denmark

1 Introduction

Optimization problems with a quadratic objective and quadratic constraints form the class of quadratically constrained quadratic programs (QCQPs). This class of problems has great modelling power and applications in many areas of science and engineering; see for example [2, 15, 21] and references therein. Unfortunately, QCQPs are generally NP-hard [27] which means that, generally speaking, there exists no efficient (i.e., polynomial time) algorithm for solving such problems. However, many problems in the class of QCQPs can be solved efficiently, *e.g.*, if the problem is convex or if the problem structure is favorable in some way. One approach to solving general QCQPs is convex relaxation which provides a lower bound on the optimal value of the original problem. Moreover, in some cases it is possible to extract a globally optimal solution to the QCQP from the solution to its relaxation. In this case we say that the relaxation is *exact*.

Our focus in this paper is *a priori* guarantees of exactness for the so-called *Shor* relaxation [24] of a QCQP, *i.e.*, guaranteeing exactness without actually solving the relaxation which is a semidefinite program (SDP). From a practical point of view, the a priori knowledge that a single problem instance will have an exact relaxation may not seem interesting (after all, this can be checked after solving the relaxation problem). However, exactness guarantees are useful, since they provide some insight into which classes of QCQPs can be solved to global optimality via its Shor relaxation. In addition, it provides a natural way to extend the exactness guarantee to a family of similar problem instances.

The question of when a relaxation of a QCQP is exact has been studied for various QCQPs in the literature. Especially trust-region type QCQPs, where the constraints are few and structured, have received a lot of attention; see [7] for a survey. Results regarding the exactness of the SDP relaxation also exist for some restricted classes of structured QCQPs with an arbitrary (but finite) number of variables and constraints. This paper extends this line of research and builds on the theory developed in [5] and [8]. Both of these papers propose sufficient conditions for the SDP relaxation to be exact, and the conditions are based on the problem structure and data.

We will consider (nonconvex) QCQPs of the form

$$\begin{array}{ll} \text{minimize} & x^H A_0 x \\ \text{subject to} & x^H A_k x + \alpha_k \leq 0, \quad k = 1, \ldots, m, \end{array} \tag{T-QCQP}$$

where $x \in \mathbb{C}^n$ is the variable, $\{A_k\}_{k=0}^m \in \mathcal{H}^n$ and $\{\alpha_k\}_{k=1}^m \in \mathbb{R}$ are the problem data, and we will restrict our attention to problems for which the aggregate sparsity pattern of the matrices $\{A_k\}_{k=0}^m$ is a forest. We call such problems *homogeneous quadratically constrined quadratic programs with forest structure*. Here, $\mathbb{R}$ is the field of real numbers, $\mathbb{C}$ is the field of complex numbers, $\mathcal{H}^n$ denotes the set of Hermitian matrices of order n, and x^H denotes the Hermitian transpose of x. We will present everything in the complex domain and note that all theory developed also holds in the real domain. When necessary, we will make a distinction between the two cases. To simplify exposition, we present our main result for QCQPs with tree structure; in Section 5 we show how the theory is readily extended to QCQPs with forest structure.

Complex-valued QCQPs can be reformulated as equivalent real-valued QCQPs via a simple transformation (see, *e.g.*, [6]). However, the complex-valued QCQP sometimes has the advantage that its relaxation can be strengthened by adding valid inequalities [9]. Moreover, the tree structure that we rely on for the conditions in this paper is generally not preserved by the transformation to the real domain.

Bose *et al.* [5] consider homogeneous complex-valued QCQPs arising from acyclic graphs, which corresponds to problems of the form (T-QCQP), and derive sufficient conditions for a problem to be solvable in polynomial time (via its SDP relaxation). Their conditions require that the point sets $\{[A_0]_{ij}, [A_1]_{ij}, \ldots, [A_m]_{ij}\}$, defined for off-diagonal indices $i \neq j$, are contained in a halfspace defined by a line that passes through the origin in the complex plane. This is closely related to the conditions derived in this paper; we will discuss the connection to our conditions, which are less conservative, in Section 5.

The main inspiration for this paper comes from a recent paper by Burer and Ye [8] in which they consider (non-homogeneous) real-valued diagonal QCQP problems, *i.e.*, problems with linear terms, but where all matrices are diagonal.

The authors prove that the infeasibility of n feasibility systems is a sufficient condition for the SDP relaxation of such problems to be exact. Moreover, as they show in the paper, it can also be applied to general QCQPs by means of lifting.

In this paper, we extend the feasibility systems of Burer and Ye [8] to homogeneous QCQPs with tree structure, and we present a sufficient condition for the relaxation of these to be exact. Moreover, if this sufficient condition holds for a given problem, we propose a robustness analysis that reveals how much the problem data can be perturbed in a given way without violating the conditions. This is based on ideas from robust optimization, and it provides a perturbation set that characterizes a family of problems for which we can guarantee an exact relaxation. We call this set a *region of exactness*.

One particular problem that can be formulated as a QCQP is the alternating current optimal power flow (ACOPF) problem. The goal is to find an optimal power dispatch for a given network topology. Distribution networks are usually radial (*i.e.*, there are no cycles) which means that the resulting QCQP has tree structure for these topologies. One of the motivations behind this paper is the observation that the SDP relaxation of the ACOPF problem usually is exact for radial (acyclic) networks [14]. In the literature, it has been proven that there exist exact convex relaxations for different practical assumptions on the network [5, 22, 19, 14, 13]. However, in practice, it is difficult to find an assumption that is valid for all cases of interest. Therefore, it is of interest to have several conditions that cover different cases and to understand the cases that are not covered.

The ACOPF problem is generally solved periodically. The problem data change from one problem instance to the next, but it is typically only a subset of $\{\alpha_k\}_{k=1}^m$ in the QCQP formulation that changes. As we will see, the first condition for exact relaxation in this paper does not depend on $\{\alpha_k\}_{k=1}^m$, so if an ACOPF problem satisfies the condition, we can extend the guarantee of exactness to a family of similar problem instances with arbitrary $\{\alpha_k\}_{k=1}^m$. Moreover, the problem data $\{A_k\}_{k=0}^m$ in the ACOPF depends on the network topology and some physical properties, both of which vary but typically do not vary too much. With the region of exactness approach presented in Section 3, we aim to

guarantee exactness of the SDP relaxation for problem instances within some region of a nominal problem instance.

Related Work

The SDP relaxation is exact if there exists a rank-1 solution. Thus, exactness can be guaranteed if, *e.g.*, it is possible to show that all solutions have rank at most 1. We will denote the minimum rank of optimal solutions of the SDP relaxation by $r^\star$. For real-valued QCQP problems, Pataki [23] and Barvinok [3] proved that

$$r^\star \leq \left\lfloor \frac{\sqrt{8m+1}-1}{2} \right\rfloor,$$

where m is the number of constraints in the problems[1]. This guarantees that for problems with less than three constraints, the SDP relaxation has a rank-1 solution. The analog bound for complex QCQP problems is $r^\star \leq \lfloor m \rfloor$ [20, Theorem 5.1].

For sparse real-valued QCQPs (with $x \in \mathbb{R}^n$ in (T-QCQP)) we have the bound $r^\star \leq \operatorname{tw}(\mathcal{G}) + 1$ where $\operatorname{tw}(\mathcal{G})$ denotes the treewidth of the sparsity graph $\mathcal{G}$ [18]. Since we limit our attention to problems whose aggregate sparsity pattern is a forest, we have $tw(\mathcal{G}) = 1$, and hence $r^\star \leq 2$. Thus, the difference between an exact and inexact relaxation is essentially whether the relaxation has a rank-1 or a rank-2 solution; the same is true for complex-valued QCQPs due to the tree structure and a result on minimum-rank positive semidefinite completion [12]. However, this bound never guarantees exactness, and it does not take the problem data into account. A homogeneous complex-valued QCQP with Toeplitz-Hermitian structure (*i.e.*, all matrices are Toeplitz-Hermitian), can be solved in polynomial time [17]—note that the SDP relaxation does not always yield a rank-1 solution in this case, but there exists a procedure to find one with the same objective value. In a recent paper, Wang and Kılınç-Karzan [28] suggest a framework for studying exactness of the SDP relaxation of a general QCQP (with linear terms and any structure for the matrices) by considering

[1] These results are for equality constrained SDPs, but they can easily be extended to inequality constrained problems, because the introduction of slack variables does not change the number of constraints.

the feasible space of the dual of the relaxation. Under the assumption that the dual feasible space is polyhedral, they provide conditions on the faces of this to guarantee exactness of the relaxation. The object that they study is similar to the one that is studied in this paper, but we assume more structure and therefore get a different set of conditions that can be checked in polynomial time. In a related line of research, Cifuentes *et al.* [11] study the geometry of the region of objective functions for which the SDP relaxation is exact.

The concept of robust feasibility presented in Section 3 is related to the concept of SDP stability discussed in [10]. In that paper, Cifuentes *et al.* consider a parameterized homogeneous QCQP and perturb away from a nominal value where the SDP relaxation is exact. The authors present conditions for the SDP relaxation to remain exact for a neighborhood around the nominal value. Their motivation for considering this problem is parameter estimation problems with noisy observations. Their conditions guarantee that the solution of the SDP relaxation is the maximum likelihood estimator in low noise settings. We assume tree structure of the matrices in this paper and provide a more explicit characterization of a *region of exactness* which is defined as a set of problem instances for which the relaxation remains exact.

A very recent paper by Azuma *et al.* [1] studies QCQPs with forest structure, and they present results that are similar in nature to some of the results presented in this paper. In particular, the main result [1, Lemma 3.5] is similar to Theorem 1 in this paper.

Contributions

Our contributions can be summarized as follows: (i) we propose a new, sufficient condition for the exactness of the SDP relaxation of (T-QCQP) when the sparsity graph is a tree or a forest; (ii) as an extension, we propose a robust exactness condition that allows us to certify exactness for a family of nearby problem instances; (iii) when the exactness condition does not hold, we propose a restricted exactness condition that allows us to guarantee exactness for a restricted set of problems.

Outline

The paper is organized as follows. We end this section by introducing the notation and terminology used throughout the paper. In Section 2, we introduce a set of feasibility systems and a sufficient condition for an exact relaxation of (T-QCQP). In Section 3, we extend this sufficient condition to a family of problems that are similar to the nominal problem (T-QCQP) in that the data matrices $\{A_k\}_{k=0}^m$ are allowed to vary within some set. In Section 4, we consider a restricted exactness condition that applies to problems that have an exact relaxation without satisfying our sufficient condition for it to be exact. In Section 5, we discuss how to extend our condition to forest-structured QCQPs and relate our condition to those proposed in [5] and in [8]. We end the paper with conclusions in Section 6.

Notation and Terminology

Define the linear operator $\mathcal{A} : \mathcal{H}^n \to \mathbb{R}^m$ by $\mathcal{A}(X) = (A_1 \bullet X, A_2 \bullet X, \ldots, A_m \bullet X)$, where $A_0 \bullet X = \mathbf{tr}(A_0^H X)$ denotes the (trace) inner product of A_0 and X. The adjoint operator $\mathcal{A}^* : \mathbb{R}^m \to \mathcal{H}^n$ is given by $\mathcal{A}^*(\lambda) = \sum_{k=1}^m \lambda_k A_k$. For convenience, we define $Y(\lambda) = A_0 + \mathcal{A}^*(\lambda)$, since this will play a significant role in our derivations. Let $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_m)$ denote a column vector. For the problem (T-QCQP), we consider the *Shor semidefinite programming relaxation* and its dual:

$$\begin{array}{cccccc} & \text{minimize} & A_0 \bullet X & & \text{maximize} & \lambda^T \alpha \\ \text{(R)} & \text{subject to} & \mathcal{A}(X) + \alpha \preceq_{\mathbb{R}^m_+} 0 & \qquad & \text{subject to} & Y(\lambda) \succeq_{\mathcal{H}^n_+} 0 \qquad \text{(RD)} \\ & & X \succeq_{\mathcal{H}^n_+} 0 & & & \lambda \succeq_{\mathbb{R}^m_+} 0. \end{array}$$

where $X \in \mathcal{H}^n$ is the variable in the primal problem (R) and λ is the variable in the dual problem (RD). The conic inequality constraint $X \succeq_{\mathcal{H}^n_+} 0$ denotes that X must be Hermitian positive semidefinite, *i.e.*, $z^H X z \geq 0$ for all $z \in \mathbb{C}^n$, and $\lambda \succeq_{\mathbb{R}^m_+} 0$ denotes that $\lambda_k \geq 0$ for $k = 1, \ldots, m$. We denote the feasible sets of

(R) and (RD) by

$$\mathcal{F} = \left\{ X : X \succeq_{\mathcal{H}_+^n} 0,\ \mathcal{A}(X) + \alpha \preceq_{\mathbb{R}_+^m} 0 \right\}, \quad \Omega = \left\{ \lambda : \lambda \succeq_{\mathbb{R}_+^m} 0,\ Y(\lambda) \succeq_{\mathcal{H}_+^n} 0 \right\}.$$

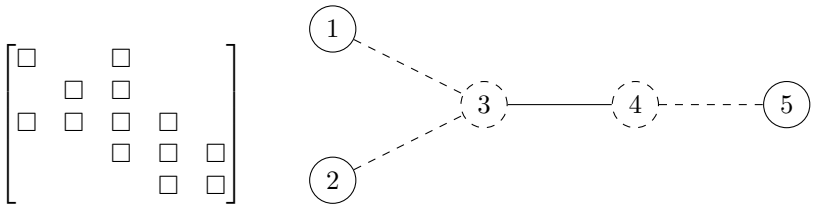
We define the matrices $E_{ij} = \frac{1}{2}(e_i e_j^T + e_j e_i^T)$ and $\bar{E}_{ij} = \frac{1}{2\imath}(e_i e_j^T - e_j e_i^T)$, where e_k denotes the kth canonical vector in $\mathbb{R}^n$ and $\imath = \sqrt{-1}$ denotes the imaginary unit. Then, for a Hermitian matrix $M = P + \imath Q$ we have $E_{ij} \bullet M = P_{ij}$ and $\bar{E}_{ij} \bullet M = Q_{ij}$. Let $\mathbf{0}_n$ denote an $n \times n$ matrix with zero in all entries.

We define the sparsity graph associated with (T-QCQP), denoted $\mathcal{G}(\mathcal{V}, \mathcal{E})$, to have vertex set $\mathcal{V} = \{1, 2, \ldots, n\}$ and an edge between vertices i and j if and only if $[A_k]_{ij} \neq 0$ for some k, *i.e.*, the graph is determined by the aggregate sparsity pattern of the matrices. In particular, if there is a nonzero in the ijth element of one of the matrices for $i < j$ (in the lower triangle), then there is an edge between vertices i and j in $\mathcal{G}(\mathcal{V}, \mathcal{E})$. In other words, let $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ denote the edge set, then

$$(i, j) \in \mathcal{E} \iff i > j \ \wedge\ \exists k \in \{0, 1, \ldots, m\} : [A_k]_{ij} \neq 0.$$

We will assume that for each $i \in \mathcal{V}$, there exists a $k \in \{0, 1, \ldots, m\}$ such that $[A_k]_{ii} \neq 0$. Note that $\mathcal{G}(\mathcal{V}, \mathcal{E})$ is a simple undirected graph. As mentioned previously, we limit our attention to problems where $\mathcal{G}(\mathcal{V}, \mathcal{E})$ is a tree; in Section 5, we discuss how to handle problems where $\mathcal{G}(\mathcal{V}, \mathcal{E})$ is a forest. For our conditions, the leaves of the graph and the non-leaf edges are important, so we proceed to define these formally. A *leaf* is a vertex that is only connected to one other vertex; a leaf is also called a simplicial node. Denote the set of leaves by $\mathcal{V}_1 = \{i \in \mathcal{V} : \deg(i) = 1\}$, where $\deg(i)$ denotes the degree of vertex i. Note that we use the term tree to mean an unrooted tree, so any vertex with degree one is a leaf. As we often associate a vertex or an edge with its corresponding matrix entry, it is convenient to define the matrix entries of the leaves as $\mathcal{L} = \{(i, i) : i \in \mathcal{V}_1\}$. Similarly, we define the set of diagonal entries as $\mathcal{D} = \{(i, i) : i \in \mathcal{V}\}$. An edge is a *non-leaf edge* if neither of the vertices that it connects is a leaf. We denote the set of non-leaf edges by $\mathcal{E}_{\text{nl}} = \{(i, j) \in \mathcal{E} : i \notin \mathcal{V}_1, j \notin \mathcal{V}_1\}$. Figure 1 shows an example of a matrix with tree structure and its sparsity graph with the leaves and non-leaf edges

highlighted. In the graph in Figure 1b, vertices correspond to diagonal elements and edges correspond do off-diagonal elements.



(a) Sparsity pattern. Nonzero elements are denoted by X while zero entries are blank.

(b) Graph of the sparsity pattern. The leaves are $\mathcal{V}_{\mathrm{l}} = \{1, 2, 5\}$ and marked by solid circles. The non-leaf edges are $\mathcal{E}_{\mathrm{nl}} = \{(3, 4)\}$ and marked by solid lines.

Figure 1: A sparsity pattern and its graph.

2 Feasibility Systems

In this section, we define a set of *feasibility systems* derived from the dual feasible set. We then use these feasibility systems to state a sufficient condition for the exactness of the SDP relaxation.

Throughout the paper, we will assume that strong duality holds for the SDP relaxation (R) and its dual (RD); this can be ensured by a constraint qualification such as Slater's condition. Let $X^\star$ denote a solution to (R) and let $\lambda^\star$ denote a solution to (RD). This means that $Y(\lambda^\star) \bullet X^\star = 0$, and hence $Y(\lambda^\star)X^\star = 0$ since both matrices are positive semidefinite. In turn, we have that

$$\mathbf{rank}\, X^\star + \mathbf{rank}\, Y(\lambda^\star) \leq n. \tag{1}$$

This is *Sylvester's rank inequality*. We will use this together with a lower bound on $\mathbf{rank}\, Y(\lambda^\star)$ to introduce an upper bound on $\mathbf{rank}\, X^\star$. For a lower bound on $\mathbf{rank}\, Y(\lambda^\star)$, we use the following proposition.

Proposition 1. *Suppose $H \in \mathcal{H}^n_+$. If the sparsity graph associated with H is a connected tree, then the rank of H is at least $n - 1$.*

Proof. This is a reformulation of [25, Theorem 3.4] and the proof can be found therein. □

We will refer to a matrix as having *connected tree structure* if its sparsity graph is a connected tree, *i.e.*, an undirected graph with no cycles where all vertices are connected by a path. A solution $\lambda^\star$ to (RD) is dual feasible, and hence $Y(\lambda^\star) \succeq_{\mathcal{H}_+^n} 0$ which implies that we have a lower bound on the rank of $Y(\lambda^\star)$ if it has connected tree structure. One way to ensure that $Y(\lambda^\star)$ has connected tree structure is to require that $Y(\lambda)$ has connected tree structure for all $\lambda \in \Omega$. This is equivalent to requiring that the following feasibility systems are all infeasible.

Definition 1 (Feasibility systems)**.** *For each $(i,j) \in \mathcal{E} \cup \mathcal{D}$, we define a* feasibility system

$$\exists \lambda \in \Omega : E_{ij} \bullet Y(\lambda) = 0,\ \bar{E}_{ij} \bullet Y(\lambda) = 0, \tag{FS$_{ij}$}$$

and we associate with this system a Boolean variable

$$f_{ij} = \begin{cases} 1 & \text{if (FS}_{ij}\text{) is feasible} \\ 0 & \text{if (FS}_{ij}\text{) is infeasible.} \end{cases}$$

Geometrically, each feasibility system represents the intersection of the dual feasible set with two hyperplanes. For real-valued problems of the form (T-QCQP), we only need a single equality condition, because $\bar{E}_{ij} \bullet Y(\lambda) = 0$ is trivially satisfied when $Y(\lambda)$ is symmetric. The feasibility systems are equivalent to SOCPs since the presence of tree structure implies that Ω is second-order cone representable [26].

For problems with tree structure there are $2n-1$ feasibility systems of the form (FS$_{ij}$) (n associated with diagonal elements and $n-1$ associated with off-diagonal elements). However, since $Y(\lambda) \succeq_{\mathcal{H}_+^n} 0$ for all $\lambda \in \Omega$, it is sufficient to consider the feasibility systems associated with the edges in the sparsity graph, because the presence of an off-diagonal nonzero implies that the corresponding diagonal elements must be nonzero as well.

Theorem 1. *If* (FS$_{ij}$) *is infeasible for all $(i,j) \in \mathcal{E}$, i.e., $\sum_{(i,j)\in\mathcal{E}} f_{ij} = 0$, then the SDP relaxation (R) of* (T-QCQP) *is exact.*

Proof. Suppose all edge feasibility systems are infeasible. All diagonal feasibility systems must also be infeasible because the constraint $Y(\lambda) \succeq_{\mathcal{H}_+^n} 0$ implies that the ith column and row of $Y(\lambda)$ must be zero if $Y(\lambda)_{ii} = 0$. Thus, all feasible dual variables $Y(\lambda)$ must have connected tree structure which, in turn, implies that $\mathbf{rank}\, Y(\lambda^\star) \geq n-1$ by Proposition 1. From Sylvester's rank inequality (1) we have that

$$\mathbf{rank}\, X^\star + \mathbf{rank}\, Y(\lambda^\star) \leq n \iff \mathbf{rank}\, X^\star \leq n - \mathbf{rank}\, Y(\lambda^\star) \leq n-(n-1) = 1,$$

and hence the SDP relaxation must be exact. □

Theorem 1 requires that all edge feasibility systems are infeasible as this is sufficient to ensure that $[Y(\lambda)]_{ij} \neq 0$ for all $\lambda \in \Omega$ for all $(i,j) \in \mathcal{E}$. For the next theorem, we will need the following result which is a generalization of Proposition 1.

Proposition 2. *A matrix $H \in \mathcal{H}_+^n$ has rank at least $n-1$ if its sparsity graph has n vertices and is a forest with a single connected tree and isolated nodes otherwise.*

Proof. Denote the number of isolated nodes by d. The matrix H has a block diagonal structure with d single elements on the diagonal and a block of size $n-d$, which we denote B. The matrix B has rank at least $n-d-1$ by Sylvester's criterion (all principal submatrices of a positive semidefinite matrix are positive semidefinite) and Proposition 1. The full matrix H up to a permutation is equal to

$$\begin{bmatrix} B & 0 \\ 0 & D \end{bmatrix}$$

where D is diagonal matrix of order d with positive diagonal elements. From this we see that

$$\mathbf{rank}\, H = \mathbf{rank}\, B + \mathbf{rank}\, D = \mathbf{rank}\, B + d \geq n - d - 1 + d = n - 1.$$

□

Theorem 2. *The SDP relaxation (R) of* (T-QCQP) *is exact if the feasibility*

systems associated with non-leaf edges and leaf nodes are all infeasible, i.e., $\sum_{(i,j)\in\mathcal{E}_{\mathrm{nl}}\cup\mathcal{L}} f_{ij} = 0$.

Proof. We consider the possible graph structure of the sparsity pattern for $Y(\lambda^\star)$. Since all non-leaf edge feasibility systems are infeasible, all interior vertices are in the graph and form a connected tree. Since all leaves are also in the graph, all nodes are in the graph. We do not know if the leaves are part of the connected tree or isolated (fallen leaves), but either way, $Y(\lambda^\star)$ has the structure of Proposition 1, so we conclude that $\mathbf{rank}\, Y(\lambda^\star) \geq n-1$. This implies that $\mathbf{rank}\, X^\star \leq 1$ as desired. □

We will refer to the feasibility systems associated with leaf nodes and non-leaf edges as the *essential feasibility systems*, since the infeasibility of this subset of feasibility systems is a sufficient condition for the relaxation to be exact. The essential feasibility systems correspond to the set $\mathcal{L}\cup\mathcal{E}_{\mathrm{nl}}$. Theorems 1 and 2 both include $n-1$ feasibility systems to check. The relaxation of the condition from Theorem 1 to Theorem 2 takes advantage of the fact that a zero on the off-diagonal of a positive semidefinite matrix does not imply a zero on the diagonal, but the converse is true for positive semidefinite matrices. Therefore, if (FS_{ij}) is infeasible, then (FS_{ii}) and (FS_{jj}) are also infeasible. Similarly, if (FS_{ii}) and/or (FS_{jj}) is feasible, then (FS_{ij}) is also feasible, *i.e.*, $f_{ij} \geq f_{ii}$ and $f_{ij} \geq f_{jj}$.

The sparsity pattern of $Y(\lambda^\star)$ provides an easy way to confirm if a relaxation is exact without computing any eigenvalues. Indeed, the relaxation is exact if $Y(\lambda^\star)$ has nonzeros in the positions corresponding to the essential feasibility systems.

The following example demonstrates the application of Theorem 1.

Example 1. *Consider the problem*

$$\begin{array}{ll} \text{minimize} & x_1^2 + 4x_2^2 + 4x_3^2 + x_4^2 + 2x_1x_2 - 2x_2x_3 + 2x_3x_4 \\ \text{subject to} & -2x_1^2 + 2x_2x_3 \leq -3 \\ & -2x_4^2 + 2x_2x_3 \leq -3 \\ & -x_4^2 + 2x_3x_4 \leq 0, \end{array} \tag{2}$$

where $x \in \mathbb{R}^4$ is the variable. This can be written in the form of (T-QCQP) *as*

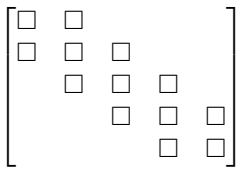
$$\begin{array}{ll} \text{minimize} & x^T A_0 x \\ \text{subject to} & x^T A_k x + \alpha_k \leq 0, \qquad k = 1, \ldots, 3, \end{array}$$

where $\alpha_1 = 3, \alpha_2 = 3, \alpha_3 = 0$ and

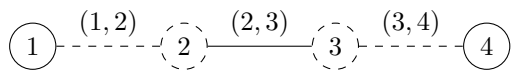
$$A_0 = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 4 & -1 & 0 \\ 0 & -1 & 4 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix}, A_1 = \begin{bmatrix} -2 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix},$$

$$A_2 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -2 \end{bmatrix}, A_3 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & -1 & -2 \end{bmatrix}.$$

The matrices A_1, A_2, and A_3 are indefinite whereas A_0 is positive definite. The sparsity pattern and the corresponding sparsity graph for problem (2) *are shown in Figure 2.*



(a) Sparsity pattern.



(b) Sparsity graph: the vertices correspond to diagonal entries and edges correspond to off-diagonal entries. The solid lines correspond to the essential feasibility systems.

Figure 2: Sparsity pattern and sparsity graph for problem (2).

The edge set of interest in Theorem 1 is

$$\mathcal{E} = \{(1,2), (2,3), (3,4)\}.$$

The leaves and non-leaf edges of interest in Theorem 2 are

$$\mathcal{V}_{\mathrm{l}} = \{1, 4\}, \ \mathcal{E}_{\mathrm{nl}} = \{(2,3)\},$$

and hence the essential feasibility systems are

$$\mathcal{L} \cup \mathcal{E}_{\text{nl}} = \{(1,1),(2,3),(4,4)\}.$$

All the edge feasibility systems are infeasible in this example, and hence all the essential feasibility systems are infeasible. Theorem 1 (or 2) leads us to the conclusion that the SDP relaxation is exact. This means that we can solve (2) *in polynomial time by solving the SDP relaxation and computing a rank-1 decomposition of $X^\star$ regardless of the problem data $\alpha_1, \alpha_2, \alpha_3$.*

To illustrate the difference between the conditions of Theorems 1 and 2, we now consider a problem where $n = 3$ and the matrices $\{A_k\}_{k=0}^m$ are tridiagonal.

Example 2. *Suppose $n = 3$ and the aggregate sparsity pattern of $\{A_k\}_{k=0}^m$ is given by*

$$\begin{bmatrix} \square & \square & \\ \square & \square & \square \\ & \square & \square \end{bmatrix},$$

which corresponds to the sparsity graph in Figure 3. The condition of Theorem 1

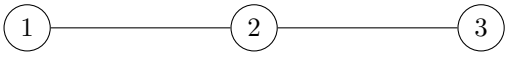


Figure 3: Sparsity graph.

is satisfied if both edges are present in the sparsity pattern of $Y(\lambda)$ for all $\lambda \in \Omega$. This implies that the sparsity graph in Figure 3 is the sparsity graph of $Y(\lambda)$ for all $\lambda \in \Omega$. The essential feasibility systems for this example are (FS_{11}) and (FS_{33}). The condition of Theorem 2 is satisfied if vertices 1 and 3 are present in the sparsity graph of $Y(\lambda)$ for all $\lambda \in \Omega$. Then the possible sparsity patterns for $Y(\lambda)$ are

$$\begin{bmatrix} \blacksquare & ? & \\ ? & ? & ? \\ & ? & \blacksquare \end{bmatrix},$$

where $\blacksquare$ denotes a non-zero element and ? denotes an arbitrary element (zero or non-zero). The possible sparsity graphs are shown in Figure 4, and each of

these guarantees that $\mathbf{rank}\, Y(\lambda) \geq 2$. *Thus, the condition of Theorem 2 is less restrictive than that of Theorem 1.*

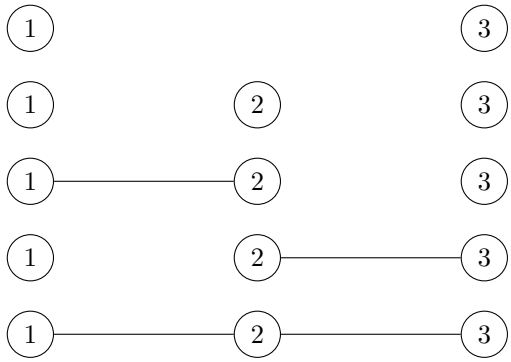


Figure 4: Possible sparsity graphs.

Alternative Feasibility Systems

For each of the feasibility system in Theorem 2, we can derive an alternative. To this end, recall the feasibility system (FS_{ij}) which seeks to find a λ such that

$$\begin{aligned} -\lambda &\preceq_{\mathbb{R}^m_+} 0 \\ -A_0 - \sum_{k=1}^{m} \lambda_k A_k &\preceq_{\mathcal{H}^n_+} 0 \\ E_{ij} \bullet (A_0 + \sum_{k=1}^{m} \lambda_k A_k) &= 0 \\ \bar{E}_{ij} \bullet (A_0 + \sum_{k=1}^{m} \lambda_k A_k) &= 0. \end{aligned} \tag{FS_{ij}}$$

A (weak) alternative to (FS_{ij}), which follows from duality, may be defined as follows.

Definition 2 (Alternative feasibility system)**.** *Given a matrix index* $(i, j) \in$

$\mathcal{D} \cup \mathcal{E}$, the corresponding alternative feasibility system is defined as

$$\begin{aligned} X &\succeq_{\mathcal{H}_+^n} 0 \\ A_0 \bullet (-X + z_1 E_{ij} + z_2 \bar{E}_{ij}) &> 0 \\ \mathcal{A}(-X + z_1 E_{ij} + z_2 \bar{E}_{ij}) &\succeq_{\mathbb{R}_+^m} 0. \end{aligned} \tag{A-FS$_{ij}$}$$

We represent the feasibility of (A-FS$_{ij}$) *by the Boolean variable*

$$\tilde{f}_{ij} = \begin{cases} 1 & \text{if (A-FS}_{ij}\text{) is feasible} \\ 0 & \text{if (A-FS}_{ij}\text{) is infeasible.} \end{cases}$$

Lemma 1. *At most one of the systems* (FS$_{ij}$) *and* (A-FS$_{ij}$) *is feasible, i.e.,* $f_{ij} + \tilde{f}_{ij} \leq 1$.

Proof. This follows from weak duality. □

Following this lemma, we can state a theorem that is similar to Theorem 2.

Theorem 3. *If* (A-FS$_{ij}$) *is feasible for all essential feasibility systems, i.e.,* $\sum_{(i,j)\in\mathcal{L}\cup\mathcal{E}_{\text{nl}}} \tilde{f}_{ij} = n - 1$, *then the SDP relaxation (R) of* (T-QCQP) *is exact.*

Proof. This follows from Theorem 2 and Lemma 1. □

Note that the condition in Theorem 3 is a sufficient condition for the condition in Theorem 2, *i.e.*, Theorem 2 is less conservative. With the added assumption that strong duality holds, the two theorems become equivalent. However, in practice, this is not necessary since one can always proceed to check (FS$_{ij}$) if (A-FS$_{ij}$) is infeasible.

3 Robust Feasibility Systems

When Theorem 3 holds, we can guarantee exactness for a family of problems with $\{A_k\}_{k=0}^m$ fixed and α arbitrary. We will now extend the guarantee to a

larger set of problems with the goal of making a robust guarantee of exactness. In other words, we would like to guarantee the exactness of the relaxation for a family of QCQPs for which $\{A_k\}_{k=0}^m$ belongs to some prescribed set. This is motivated by the uncertainty in many engineering applications where the problem data ($\{A_k\}_{k=0}^m$ and $\{\alpha_k\}_{k=1}^m$) may come from measurements or estimates and are inherently uncertain. Another motivation is applications where we would like to solve many similar optimization problems where the data are a little different each time depending on, for example, outside conditions.

We note that our objective is not to solve a robust optimization problem, *i.e.*, optimizing the worst-case objective for problem data within the uncertainty set. Instead, we wish to certify relaxation exactness for all problem data within some uncertainty set.

The starting point for robust feasibility is that we have a *nominal problem* for which Theorem 3 holds.

Assumption 1. *Given a problem of the form* (T-QCQP) *with data* $\{\hat{A}_k\}_{k=0}^m$ *and* $\{\hat{\alpha}_k\}_{k=1}^m$*, assume that Theorem 3 holds.*

Any problem satisfying Assumption 1 may be used to compute a region of exactness as we explain next.

Region of Exactness

Suppose that Assumption 1 holds for a nominal problem with nominal data $\{\hat{A}_k\}_{k=0}^m$, *i.e.*, all essential alternative feasibility systems are feasible, and hence the semidefinite relaxation will be exact regardless of $\{\alpha_k\}_{k=1}^m$. For each nominal matrix $\hat{A}_k$, we define a perturbation set $Z_k(\rho) \subseteq \mathcal{H}^n$ where $\rho \in \mathbb{R}_+$ is a parameter that controls the size of the set, and $Z_k(\rho)$ only contains matrices with a sparsity pattern contained in that of the nominal matrices (tree-structure). We allow for different perturbation sets for the individual nominal matrices, but we limit our attention to perturbation sets that are all parameterized by the same

parameter, ρ, and are monotonically increasing in this parameter, *i.e.*,

$$\zeta < \eta \implies Z_k(\zeta) \subset Z_k(\eta).$$

For a given ρ, we can describe a set of matrices by

$$\{\hat{A}_k + U_k\}_{U_k \in Z_k(\rho)}.$$

The choice of perturbation set $Z_k(\rho)$ should reflect any knowledge of the uncertainty of the entries of $\hat{A}_k$. Some special cases are:

- $Z_k(\rho) = \{\mathbf{0}_n\}$ if the data in $\hat{A}_k$ are not subject to any uncertainty.
- $Z_k(\rho) = \{\eta E_{st} : |\eta| \leq \rho\}$ if $\mathrm{Re}([\hat{A}_k]_{st})$ is uncertain but the rest of $\hat{A}_k$ is not subject to any uncertainty.
- $Z_k(\rho) = \{\eta T : |\eta| \leq \rho\}$, where T denotes a matrix of ones whose sparsity pattern matches that of the matrices $\{\hat{A}_k\}_{k=0}^m$, *i.e.*, it has ones on the diagonal and on the off-diagonal elements that have an edge in $\mathcal{G}$(T-QCQP). This would reflect that all entries of $\hat{A}_k$ are subject to the same amount of uncertainty.

Analogously to the alternative feasibility systems for a nominal problem, we define the robust feasibility systems for the set of perturbed problems, *i.e.*, problems with data from the perturbed sets.

Definition 3 (Robust feasibility systems)**.** *Given perturbation sets* $\{Z_k(\rho)\}_{k=0}^m$*, the robust feasibility system for* $(i,j) \in \mathcal{D} \cup \mathcal{E}$ *checks if there exists an* X *such that*

$$\begin{aligned} &X \succeq_{\mathcal{H}_+^n} 0 \\ &(\hat{A}_0 + U_0) \bullet (-X + z_1 E_{ij} + z_2 \bar{E}_{ij}) > 0, \ \forall U_0 \in Z_0(\rho) \\ &(\hat{A}_k + U_k) \bullet (-X + z_1 E_{ij} + z_2 \bar{E}_{ij}) \geq 0, \ \forall U_k \in Z_k(\rho), \ k = 1, \ldots, m. \end{aligned} \tag{R-FS$_{ij}(\rho)$}$$

We represent the feasibility of (R-FS$_{ij}(\rho)$) *by the Boolean variable*

$$r_{ij}(\rho) = \begin{cases} 1 & \textit{if } (\text{R-FS}_{ij}(\rho)) \textit{ is feasible} \\ 0 & \textit{if } (\text{R-FS}_{ij}(\rho)) \textit{ is infeasible.} \end{cases}$$

The label of the system and input of the Boolean variable emphasizes the dependency ρ. Note that if we choose uncertainty sets that collapse to zero, *i.e.*, $Z_k(0) = \{\mathbf{0}_n\}$ for all k, then $r_{ij}(0) = 1$ for all (i, j) by Assumption 1. Each of the robust feasibility systems (R-FS$_{ij}(\rho)$) include $m + 1$ linear, and possibly semi-infinite, inequalities.

The following extension of Theorem 3 establishes that the relaxation will be exact for all perturbed problems if all essential robust feasibility systems (R-FS$_{ij}(\rho)$) are feasible.

Theorem 4. *Let $Z_k(\rho)$ be monotonely increasing in ρ, i.e., $\zeta < \eta \implies Z_k(\zeta) \subset Z_k(\eta)$. If the essential robust feasibility systems* (R-FS$_{ij}(\rho)$) *are feasible for some $\rho \geq 0$, then the SDP relaxation is exact for all problems with matrices from the set $\{\hat{A}_k + U_k\}_{k=0}^m$, where $U_k \in Z_k(\rho)$.*

Proof. Suppose that all essential robust feasibility systems (R-FS$_{ij}(\rho)$) are feasible for a given ρ. For any problem with matrices from the sets $A_k \in \{\hat{A}_k + U_k\}_{U_k \in Z_k(\rho)}$, the alternative feasibility system contains a subset of the inequalities of the the robust feasibility system, so the alternative feasibility system is feasible. Hence, Theorem 3 guarantees exactness for any problem with matrices $A_k \in \{\hat{A}_k + U_k\}_{U_k \in Z_k(\rho)}$. □

To guarantee exactness for all perturbed problems, we need to check the feasibility of (R-FS$_{ij}(\rho)$). Each of the sets of inequalities in (R-FS$_{ij}(\rho)$) can be represented by its robust counterpart [4]. We can express this as

$$\inf_{U_l \in Z_l(\rho)} \Big((\hat{A}_l + U_l) \bullet (-X + z_1 E_{ij} + z_2 \bar{E}_{ij}) \Big) \geq 0. \tag{3}$$

Whether (3) can be expressed in a way that makes (R-FS$_{ij}(\rho)$) tractable depends on the perturbation sets. An important observation is that the infimum in (3) is

taken over an expression that is linear in X, and this means that we can handle many perturbation sets that are affinely parameterized [4]. To illustrate this, we consider interval uncertainty in Example 3.

The robust feasibility systems in Theorem 4 depend on the parameter ρ, and hence an obvious question is how large can we make ρ while maintaining feasibility. In other words, we are interested in the largest possible ρ for which Theorem 4 holds in order to guarantee exactness for as large a family of problems as possible.

Recall that $r_{ij}(\rho) = 1$ when (R-FS$_{ij}(\rho)$) is feasible. Since we require the perturbation sets to be increasing in ρ, we have that $r_{ij}(\zeta) \leq r_{ij}(\eta)$ for $\zeta < \eta$. It follows that $r_{ij}(\rho) = 1$ for $(i,j) \in \mathcal{L} \cup \mathcal{E}_{\mathrm{nl}}$ if ρ is less than $\sup\{\rho : r_{ij}(\rho) = 1\}$ which may be computed using bisection. Similarly, the largest ρ for which Theorem 4 holds, denoted $\rho^\star$, may be expressed as

$$\rho^\star = \sup\{\rho : r_{ij}(\rho) = 1 \quad \forall (i,j) \in \mathcal{L} \cup \mathcal{E}_{\mathrm{nl}}\}.$$

Thus, we can guarantee exactness for all problems with matrices from the set $\{\hat{A}_k + U_k\}_{k=0}^m$, where $U_k \in Z_k(\rho)$ with $\rho < \rho^\star$, and we will refer to the perturbation sets $\{Z_k(\rho^\star)\}_{k=0}^m$ as a *region of exactness*.

The following example demonstrates how to derive the robust counterpart when we have a single uncertain element in a matrix and how to compute the largest interval.

Example 3 (continuation of Example 1)**.** *We will consider a problem with interval uncertainty on a single element. We start by deriving the robust counterpart. Suppose we have interval uncertainty on the real part of element st of matrix l, i.e.,* $Z_l(\rho) = \{\eta E_{st} : |\eta| \leq \rho\}$*. We need to distinguish between two cases: when*

$i = s \wedge j = t$ and when $i \neq s \vee j \neq t$. When $i = s \wedge j = t$, we can express (3) *as*

$$\begin{aligned}
&\inf_{\{\eta\in\mathbb{R}\,:\,|\eta|\leq\rho\}} \Big((\hat{A}_l + \eta E_{st}) \bullet (-X + z_1 E_{st} + z_2 \bar{E}_{st})\Big) \geq 0 \\
\Leftrightarrow\ &\inf_{\{\eta\in\mathbb{R}\,:\,|\eta|\leq\rho\}} \Big\{\eta(z_1 - X_{st}) + \hat{A}_l \bullet (-X + z_1 E_{st} + z_2 \bar{E}_{st})\Big\} \geq 0 \\
\Leftrightarrow\ &-\rho\,|z_1 - X_{st}| + \hat{A}_l \bullet (-X + z_1 E_{st} + z_2 \bar{E}_{st}) \geq 0 \\
\Leftrightarrow\ &\begin{cases} \rho u \leq \hat{A}_l \bullet (-X + z_1 E_{st} + z_2 \bar{E}_{st}) \\ -u \leq z_1 - X_{st} \leq u, \end{cases}
\end{aligned} \tag{4}$$

where X, u, z_1, z_2 are the variables. Similarly, when $i \neq s \vee j \neq t$, we can express (3) *as*

$$\begin{aligned}
&\inf_{\{\eta\in\mathbb{R}\,:\,|\eta|\leq\rho\}} \Big((\hat{A}_l + \eta E_{st}) \bullet (-X + z_1 E_{ij} + z_2 \bar{E}_{ij})\Big) \geq 0 \\
\Leftrightarrow\ &\inf_{\{\eta\in\mathbb{R}\,:\,|\eta|\leq\rho\}} \Big\{-\eta X_{st} + \hat{A}_l \bullet (-X + z_1 E_{ij} + z_2 \bar{E}_{ij})\Big\} \geq 0 \\
\Leftrightarrow\ &-\rho\,|X_{st}| + \hat{A}_l \bullet (-X + z_1 E_{ij} + z_2 \bar{E}_{ij}) \geq 0 \\
\Leftrightarrow\ &\begin{cases} \rho u \leq \hat{A}_l \bullet (-X + z_1 E_{ij} + z_2 \bar{E}_{ij}) \\ -u \leq X_{st} \leq u. \end{cases}
\end{aligned} \tag{5}$$

Hence, the lth set of inequalities in (R-FS$_{ij}(\rho)$) *can be replaced by three inequalities; it is replaced by* (4) *in the st-feasibility system and by* (5) *in the rest.*

The problem in Example 1 satisfies Assumption 1, i.e., all essential alternative feasibility systems are feasible. Now suppose that the entry in the third row and fourth column (and fourth row and third column) of A_3 is uncertain while the rest of the entries are certain, i.e., $[A_3]_{34} = [A_3]_{43} = -1 + \eta$, $|\eta| \leq \rho$. We wish to compute the largest interval such that the SDP relaxation remains exact for all problems where $[A_3]_{34} = [A_3]_{43}$ is within this interval. This corresponds to choosing the perturbation sets $Z_0(\rho) = \{\mathbf{0}_n\}$, $Z_1(\rho) = \{\mathbf{0}_n\}$, $Z_2(\rho) = \{\mathbf{0}_n\}$, and $Z_3(\rho) = \{\eta E_{34} \ \mid \ |\eta| \leq \rho\}$ The essential robust feasibility systems are

$(i,j) \in \{(1,1),(2,3),(4,4)\}$. *We can use* (5) *to formulate these as*

$$\begin{aligned} & X \succeq_{\mathcal{S}_+^n} 0 \\ & -X_{11} - 2X_{21} - 4X_{22} + 2X_{23} - 4X_{33} - 2X_{34} - X_{44} + z_1[A_0]_{ij} > 0 \\ & 2X_{11} - 2X_{23} + z_1[A_1]_{ij} \geq 0 \\ & -2X_{23} + 2X_{44} + z_1[A_2]_{ij} \geq 0 \\ & 2\rho u \leq 2X_{34} + 2X_{44} + z_1[A_3]_{ij} \\ & -u \leq X_{34} \leq u. \end{aligned} \tag{6}$$

The data $[A_0]_{ij}, [A_1]_{ij}, [A_2]_{ij}, [A_3]_{ij}$ *depend on which feasibility system we consider:*

$$\begin{aligned} &[A_0]_{11} = 1, [A_1]_{11} = -2, [A_2]_{11} = 0, [A_3]_{11} = 0, \\ &[A_0]_{23} = -1, [A_1]_{23} = 1, [A_2]_{23} = 1, [A_3]_{23} = 0, \\ &[A_0]_{44} = 1, [A_1]_{44} = 0, [A_2]_{44} = -2, [A_3]_{44} = 0. \end{aligned}$$

Using bisection for the essential robust feasibility systems, we obtain the values in Table 1. We see that $\rho^\star = 1$*, so the SDP relaxation will be exact if* $[A_3]_{34} \in (-2;0)$.

i	j	$\max\{\rho : r_{ij}(\rho) = 1\}$
1	1	∞
3	2	∞
4	4	1

Table 1: Maximal radius in the essential robust feasibility systems for uncertainty in $[A_3]_{34}$.

4 Restricted feasibility systems

Recall that the problem data $\{\alpha_k\}_{k=1}^m$ neither play a role in the feasibility systems (FS_{ij}) nor in Theorem 2; the feasibility systems only rely on $\{A_k\}_{k=0}^m$. Consequently, if Theorem 2 holds for a set of matrices $\{A_k\}_{k=0}^m$, the SDP relaxation is guaranteed to be exact for any $\{\alpha_k\}_{k=1}^m$ in the constraints, provided

that the original problem (T-QCQP) is feasible. We now address the situation where the relaxation is exact but Theorem 2 does not hold. In this case, we may still be able to guarantee exactness for a range of problems if the solution $Y(\lambda^\star)$ to the dual problem (RD) has connected tree structure. The basic idea is to introduce a restriction in the feasibility systems, so that we only consider $\lambda \in \Omega$ that are close to $\lambda^\star$.

Our starting point for the restricted feasibility systems is the following assumption.

Assumption 2. *Given a problem of the form* (T-QCQP) *with data* $\{\hat{A}_k\}_{k=0}^m$ *and* $\{\hat{\alpha}_k\}_{k=1}^m$ *assume that:*

(i) Theorem 2 does not hold, i.e., at least one essential feasibility system is feasible.

(ii) The solution $Y(\lambda^\star)$ *to the dual problem (RD) has connected tree structure.*

Now suppose that Assumption 2 holds. Then there exists a feasible λ such that $Y(\lambda)$ does not have connected tree structure, but for the given problem, $Y(\lambda^\star)$ *does* have connected tree structure. However, since Theorem 2 relies on connected tree structure of $Y(\lambda)$ for all $\lambda \in \Omega$, we have no a priori guarantee of exactness which can be attributed to the fact that Theorems 1–3 guarantee the exactness for any $\{\alpha_k\}_{k=1}^m$. However, in many applications the $\{\alpha_k\}_{k=1}^m$ of interest is restricted in some way—for example, it may be known that $\alpha \succeq_{\mathbb{R}_+^m} 0$. In this section, we try to use the information that $Y(\lambda^\star)$ has connected tree structure to guarantee exactness for $\{\alpha_k\}_{k=1}^m$ that are close to $\{\hat{\alpha}_k\}_{k=1}^m$. Unfortunately, we were not able to obtain an explicit characterization of a perturbation set defined by a neighborhood around $\{\hat{\alpha}_k\}_{k=1}^m$, but rather as a perturbation set defined by a neighborhood around a nominal, optimal dual variable $\lambda^\star$ obtained by solving the nominal problem. We now define a set of *restricted* feasibility systems.

Definition 4 (Restricted Feasibility System)**.** *Suppose Assumption 2 holds and let* $\hat{\lambda}$ *denote the solution to (RD) with the data* $\{\hat{A}_k\}_{k=0}^m$ *and* $\{\hat{\alpha}_k\}_{k=1}^m$*. Let* $\hat{\Omega}(\rho) \subseteq \Omega$ *be a restriction of the dual feasible set that is monotonely increasing*

in $\rho \in \mathbb{R}_+$ *and where* $\hat{\lambda} \in \hat{\Omega}(\rho)$ *for all* ρ. *Given* $(i,j) \in \mathcal{D} \cup \mathcal{E}$, *the* restricted feasibility system *is given by*

$$\exists \lambda \in \hat{\Omega} : E_{ij} \bullet Y(\lambda) = 0,\ \bar{E}_{ij} \bullet Y(\lambda) = 0. \qquad \text{(Res-FS}_{ij}(\rho))$$

The point $\hat{\lambda}$ is optimal for (RD) with the objective $\hat{\alpha}^T \lambda$, and the restricted feasibility systems are an indication of how much the data $\{\hat{\alpha}_k\}_{k=1}^m$ can be perturbed while still having an exact relaxation. The restriction set is increasing in ρ, and we are interested in finding the largest set that yield an exact relaxation. This is illustrated in the following example for the restriction set $\hat{\Omega}(\rho) = \{\lambda \in \Omega : \|\lambda - \hat{\lambda}\| \leq \rho\}$.

Example 4. *Consider the problem*

$$\begin{array}{ll} \text{minimize} & x_1^2 + 4x_2^2 + 4x_3^2 + x_4^2 + 2x_1x_2 - 2x_2x_3 + 2x_3x_4 \\ \text{subject to} & -2x_1^2 + 2x_2x_3 \leq -3 \\ & -2x_4^2 + 2x_2x_3 \leq -3 \\ & -x_4^2 + 2x_3x_4 \leq 0 \\ & \|x\|^2 \leq 5. \end{array}$$

This is the problem from the previous example with a norm constraint added. In the form of (T-QCQP) *it can be written as*

$$\begin{array}{ll} \underset{x \in \mathbb{R}^4}{\textit{minimize}} & x^T A_0 x \\ \textit{subject to} & x^T A_p x + r_p \leq 0, \qquad p = 1, \ldots, 4, \end{array}$$

where $r_1 = 3, r_2 = 3, r_3 = 0, r_4 = -5$ *and*

$$A_0 = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 4 & -1 & 0 \\ 0 & -1 & 4 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix}, A_1 = \begin{bmatrix} -2 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, A_2 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -2 \end{bmatrix},$$

$$A_3 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & -1 & -2 \end{bmatrix}, A_4 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

The matrices A_1*,* A_2*, and* A_3 *are indefinite while* A_0 *and* A_4 *are positive definite. Compared to the problem in Example 1, the extra constraint* $\|x\|^2 \leq 5$ *provides added freedom in the dual problem, so all feasibility systems are no longer infeasible, as was the case in Example 1. In particular, the essential feasibility systems* $(2, 3)$ *and* $(4, 4)$ *are feasible.*

*However, the solution to the SDP relaxation (R) is the same as before (***rank** $X^\star =$ 1 *and* $Y(\lambda^\star)$ *has connected tree structure), so Assumption 2 holds. Solving the dual of the relaxation, we have*

$$\hat{\lambda} = \begin{bmatrix} 3.66 \cdot 10^{-1} \\ 3.66 \cdot 10^{-1} \\ 0 \\ 0 \end{bmatrix}$$

For the systems that are now feasible, we find the smallest ρ *for which the restricted feasibility system is feasible by solving*

$$\begin{aligned} &\textit{minimize} && \rho \\ &\textit{subject to} && Y(\lambda) \succeq_{\mathcal{S}^n_+} 0 \\ &&& E_{ij} \bullet Y(\lambda) = 0 \\ &&& \lambda \succeq_{\mathbb{R}^m_+} 0 \\ &&& \|\lambda - \hat{\lambda}\| \leq \rho. \end{aligned} \tag{7}$$

We solve this for $(i,j) \in \{(2,3),(4,4)\} = \mathcal{L} \cup \mathcal{E}_{\text{nl}} \setminus \{(1,1)\}$ *and get the following radii (denoting the radius with the index for the feasibility system):*

$$\rho_{23} \approx 0.3027, \;\; \rho_{44} \approx 1.461.$$

Thus, for all λ *within a radius of* $\rho < 0.3027$ *from* $\hat{\lambda}$*, we can guarantee that the relaxation is exact. This means that all problems parameterized by* $\{\alpha_k\}_{k=1}^m$ *for which the optimal dual variables are within this ball are guaranteed to have an exact relaxation.*

The idea of the restricted feasibility systems is essentially to guarantee exactness for a limited set of problem data $\{\alpha_k\}_{k=1}^m$ instead of guaranteeing exactness for all $\{\alpha_k\}_{k=1}^m \in \mathbb{R}^m$ (where (T-QCQP) is feasible). Ideally, we would like to guarantee exactness for any $\{\alpha_k\}_{k=1}^m \in C$ with an explicit characterization of the set $C \subset \mathbb{R}^m$. Unfortunately, we do not have a way to compute this, so for the current restrictions, we must rely on an assumption that small perturbation from $\{\hat{\alpha}_k\}_{k=1}^m$ results in small perturbations from $\hat{\lambda}$.

5 Discussion

In this section, we first present an example with a problem structure for which we can guarantee the exactness of the SDP relaxation. We then outline how our theory extends to forest-structured QCQPs, and we relate our conditions to existing conditions from the literature. Finally, we discuss some future research opportunities.

Tree Objective with Diagonal Constraints

Consider a problem of the form

$$\begin{aligned} &\underset{x \in \mathbb{C}^n}{\text{minimize}} \;\; x^H T x \\ &\text{subject to} \;\; x^H D_k x + r_k \leq 0, \quad k = 1, \ldots, m, \end{aligned} \tag{8}$$

where T has connected tree structure and the matrices D_k are diagonal. The matrix T can have zeroes on the diagonal as long as the off-diagonal elements form a connected tree.

For this problem structure—regardless of the data in the matrices—the SDP relaxation is exact. To see this, we consider the feasibility systems, which take the form

$$\exists \lambda \in \Omega : E_{ij} \bullet Y(\lambda) = 0,\ \bar{E}_{ij} \bullet Y(\lambda) = 0,\ (i,j) \in \mathcal{D} \cup \mathcal{E}.$$

Since all the matrices D_k are diagonal, the conditions $E_{ij} \bullet Y(\lambda) = 0$ and $\bar{E}_{ij} \bullet Y(\lambda) = 0$ cannot both be satisfied for any edge feasibility system ($(i,j) \in \mathcal{E}$), since $T_{ij} \neq 0$. Hence, Theorem 1 holds for problem (8).

In conclusion, due to the connectedness of the objective and the sparsity of the constraints, we can guarantee the exactness of the semidefinite relaxation of (8) regardless of the data in the problem if (8) is feasible.

Forest-Structure Quadratic Programs

We consider a forest-structured QCQP to be a homogeneous quadratic program whose sparsity graph (the graph of the aggregate nonzero pattern of the matrices) is a forest, *i.e.*, it has two or more connected components, which are all trees. We briefly outline how the conditions for tree-structured QCQPs may be extended to forest-structured ones. The following argument uses two trees for the sake of presentation, but it extends to any number of trees by induction. Specifically, we consider a problem with two connected components of the form

$$\begin{aligned} \text{minimize} \quad & x^H \begin{bmatrix} T_{0,1} & 0 \\ 0 & T_{0,2} \end{bmatrix} x = x_1^H T_{0,1} x_1 + x_2^H T_{0,2} x_2 \\ \text{subject to} \quad & x^H \begin{bmatrix} T_{k,1} & 0 \\ 0 & T_{k,2} \end{bmatrix} x + \alpha_k = x_1^H T_{k,1} x_1 + x_2^H T_{k,2} x_2 + \alpha_k \leq 0, \\ & \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad k = 1, \ldots, m, \end{aligned} \tag{9}$$

where the aggregate sparsity pattern of $\{T_{k,1}\}_{k=0}^m$ and that of $\{T_{k,2}\}_{k=0}^m$ are both connected trees. Hence, the aggregate sparsity pattern of the matrices

$$\left\{ \begin{bmatrix} T_{k,1} & 0 \\ 0 & T_{k,2} \end{bmatrix} \right\}_{k=0}^{m}$$

is a forest with two trees. The semidefinite relaxation of (9) can be formulated as

$$\begin{array}{ll} \text{minimize} & T_{0,1} \bullet X_1 + T_{0,2} \bullet X_2 \\ \text{subject to} & T_{k,1} \bullet X_1 + T_{k,2} \bullet X_2 + \alpha_k \leq 0, \ k = 1, \ldots, m, \\ & X_1 \succeq_{\mathcal{H}_+^n} 0, \ X_2 \succeq_{\mathcal{H}_+^n} 0. \end{array}$$

If $X_1^\star$ and $X_2^\star$ are both rank-1 matrices, then the relaxation is exact. To guarantee this, we can apply Theorem 2 twice; once with the linear operator $Y_1(\lambda) = T_{0,1} + \sum_{k=1}^m \lambda_k T_{k,1}$, and again with the linear operator $Y_2(\lambda) = T_{0,2} + \sum_{k=2}^m \lambda_k T_{k,2}$. This results in two sets of feasibility systems:

$$\exists \lambda \in \Omega : E_{ij} \bullet Y_1(\lambda) = 0, \bar{E}_{ij} \bullet Y_1(\lambda) = 0$$

and

$$\exists \lambda \in \Omega : E_{ij} \bullet Y_2(\lambda) = 0, \bar{E}_{ij} \bullet Y_2(\lambda) = 0$$

where

$$\Omega = \{\lambda : Y_1(\lambda) \succeq 0, \ Y_2(\lambda) \succeq 0, \ \lambda \succeq 0\}.$$

If the essential feasibility systems are all infeasible for both $Y_1(\lambda)$ and $Y_2(\lambda)$, then the relaxation of (9) is exact. The robust feasibility systems can be extended to QCQPs with forest structure analogously.

Related Conditions

We now discuss some similarities with the condition proposed by Bose *et al.* [5], which applies to homogeneous QCQPs with tree structure, and the condition proposed by Burer and Ye [8], which applies to diagonal QCQPs as well as general QCQPs by means of diagonalization and lifting.

Off-Diagonally Linearly Separable The condition of [5] checks if the set of points $P_{ij} = \{[A_0]_{ij}, [A_1]_{ij}, \ldots, [A_m]_{ij}\}$ is linearly separable from the origin for all $i \neq j$ (off-diagonal elements). This means that there exists a closed halfspace, defined by a line through the origin in the complex plane, that contains P_{ij}. As mentioned in the proof in [5], this corresponds to checking that zero is not in the interior of the convex hull of P_{ij}. Expressing the convex hull in barycentric coordinates, zero is included in the convex hull of P_{ij} if the following system is feasible

$$\nu_0[A_0]_{ij} + \nu_1[A_1]_{ij} + \cdots + \nu_m[A_m]_{ij} = 0, 0 \prec \nu \prec 1, \sum_{k=0}^{m} \nu_k = 1.$$

Thus, the condition of Bose *et al.* [5] corresponds to requiring that the above system is infeasible for all off-diagonal elements in $\mathcal{E}$. Notice that since $\nu_0 > 0$, we eliminate ν_0 and define $\lambda_k = \nu_k/\nu_0$, resulting in the equivalent formulation:

$$[A_0]_{ij} + \lambda_1[A_1]_{ij} + \cdots + \lambda_m[A_m]_{ij} = 0, \lambda \succ 0. \tag{10}$$

This is essentially the feasibility system (FS_{ij}) without the constraint $Y(\lambda) \succeq_{\mathcal{H}_+^n} 0$ and excluding the boundary of the nonnegative orthant (*i.e.*, $\lambda \succ 0$ instead of $\lambda \succeq 0$. This illustrates that the condition proposed in this paper is closely related to that in [5] but generally less conservative.

The off-diagonal linearly separable condition of Bose *et al.* is cheaper to check but generally also weaker, since it requires all edges to be present in the graph. However, our condition includes only the essential feasibility systems, allowing us to ignore certain edges in the graph. Furthermore, the feasibility systems also include the constraint $Y(\lambda) \succeq_{\mathcal{H}_+^n} 0$, so even if there exist multipliers in (10) that remove an edge from the graph, those multipliers may not be feasible (*i.e.*, $Y(\lambda) \not\succeq_{\mathcal{H}_+^n} 0$).

Diagonal QCQPs and Diagonalization The diagonal QCQPs considered by Burer and Ye [8] can be formulated as a homogeneous QCQP with arrow structure—which has a graph that is a star—by introducing two inequality constraints. For this particular QCQP, the essential feasibility systems are all the diagonal elements except the first (which corresponds to the internal vertex

of the star), and these are the ones considered in [8]. Hence, in the case of real-valued diagonal QCQPs, our conditions are equivalent to those in [8]. We briefly discuss diagonal QCQPs and diagonalization.

A diagonal QCQP takes the form

$$\begin{array}{ll} \text{minimize} & x^T D_0 x + q_0^T x \\ \text{subject to} & x^T D_k x + q_k^T x + r_k \leq 0, \qquad k = 1, \ldots, m, \end{array} \tag{11}$$

where $x \in \mathbb{R}^n$ is the variable and for $p = 0, \ldots, m$, D_k is a diagonal matrix and $q_k \in \mathbb{R}^n$. This can equivalently be formulated as

$$\begin{array}{ll} \text{minimize} & \begin{bmatrix} 1 \\ x \end{bmatrix}^T \begin{bmatrix} 0 & q_0^T \\ q_0 & D_0 \end{bmatrix} \begin{bmatrix} 1 \\ x \end{bmatrix} \\ \text{subject to} & \begin{bmatrix} 1 \\ x \end{bmatrix}^T \begin{bmatrix} r_k & q_k^T \\ q_k & D_k \end{bmatrix} \begin{bmatrix} 1 \\ x \end{bmatrix} \leq 0, \qquad k = 1, \ldots, m \end{array}$$

which, in turn, can be written as

$$\begin{array}{ll} \text{minimize} & y^T A_0 y \\ \text{subject to} & y^T A_k y \leq 0, \qquad k = 1, \ldots, m+2 \end{array} \tag{12}$$

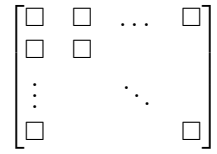
where $y \in \mathbb{R}^{n+1}$ is the variable and

$$A_k = \begin{bmatrix} r_k & q_k^T \\ q_k & D_k \end{bmatrix}, \; k = 0, \ldots, m,$$

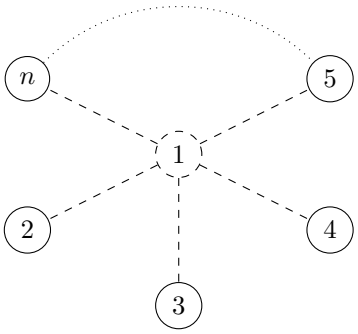
$A_{m+1} = e_1 e_1^T$, $A_{m+2} = -e_1 e_1^T$ and $b_{m+1} = b_{m+2} = 1$. Here we have introduced two inequalities for the equality $y_1 = 1$ to fit the form of (T-QCQP). Since the matrices D_k $(k = 0, \ldots, m)$ are diagonal, the graph of (12) has tree structure. In particular, the graph—which can be seen in Figure 5b—is a star, which is also mentioned in [8]. To extend the methodology to general QCQPs, Burer and Ye note that a general QCQP of the form

$$\begin{array}{ll} \text{minimize} & x^T Q_0 x + q_0^T x \\ \text{subject to} & x^T Q_k x + q_k^T x + r_k \leq 0, \qquad k = 1, \ldots, m, \end{array}$$

can be reformulated to diagonal QCQP as follows. To this end, we let $Q_k = V_k \Lambda_k V_k^T$



(a) Sparsity pattern.



(b) Graph of the sparsity pattern. Vertices correspond to diagonal entries and edges correspond to off-diagonal entries. The vertices and edges corresponding to essential feasibility systems are marked by solid lines.

Figure 5: Sparsity pattern and its graph for diagonal QCQPs.

be the eigendecomposition of Q_k, and we introduce the new variables $y_k = V_k^T x$, which results in the equivalent problem

$$\begin{array}{ll} \text{minimize} & y_0^T \Lambda_0 y_0 + q_0^T x \\ \text{subject to} & y_k^H \Lambda_k y_k + q_k^T x + r_k \leq 0,\ y_k = V_k^T x, \qquad k = 1, \ldots, m, \end{array} \tag{13}$$

which has additional variables and constraints.

The Shor relaxation of (13) is given by

$$\begin{array}{ll} \text{minimize} & \Lambda_0 \bullet Y_0 + q_0^T x \\ \text{subject to} & \Lambda_k \bullet Y_k + q_k^T x + r_k \leq 0,\ y_k = V_k^T x, \qquad k = 1, \ldots, m, \\ & \begin{bmatrix} 1 & y_k^T \\ y_k & Y_k \end{bmatrix} \succeq 0, \qquad k = 0, \ldots, m, \end{array}$$

and substituting $V_k^T x$ for y_k, we have

$$\begin{array}{ll} \text{minimize} & \Lambda_0 \bullet Y_0 + q_0^T x \\ \text{subject to} & \Lambda_k \bullet Y_k + q_k^T x + r_k \leq 0, \qquad k = 1, \ldots, m, \\ & \begin{bmatrix} 1 & x^T V_k \\ V_k^T x & Y_k \end{bmatrix} \succeq 0, \qquad k = 0, \ldots, m, \end{array}$$

where $Y_0, \ldots, Y_m \in \mathcal{S}^n$ and $x \in \mathbb{R}^n$ are the variables. It is clear from (5) that the coupling of the constraints comes from the original variable x, and that the lifting (the diagonalization) has made the problem more decoupled. Consider, in particular, the case without linear terms ($q_k = 0,\ k = 0, \ldots, m$), which will be diagonalized and relaxed to

$$\begin{array}{ll} \text{minimize} & \Lambda_0 \bullet Y_0 \\ \text{subject to} & \Lambda_k \bullet Y_k + r_k \leq 0, \qquad k = 1, \ldots, m, \\ & \begin{bmatrix} 1 & x^T V_k \\ V_k^T x & Y_k \end{bmatrix} \succeq 0, \qquad k = 0, \ldots, m. \end{array}$$

For any feasible $Y_0, \ldots, Y_m$ it is always possible to choose $x = 0$ without changing the objective or compromising feasibility. In other words, x can be eliminated from the relaxation. However, in the case where $q_k \neq 0$, the values of q_k play a significant role, and the analysis becomes more cumbersome. This example illustrates that without the presence of linear terms, the diagonalization decouples the variables.

Future Research

The results in this paper rely on the assumption that the sparsity graph associated with the QCQP of interest is a tree or a forest. A natural question to ask is therefore if similar results can be derived for other types of structure which would make it possible to construct exactness conditions for other classes of QCQPs.

Another topic of interest is techniques for tightening SDP relaxations. One such approach is to add valid cuts (additional constraints) to the relaxation (R); see,

for example, [16]. The strengthened SDP relaxation is a relaxation of the same problem as the (standard) SDP relaxation, but it is generally stronger and will be exact for a larger set of problem instances. Therefore, it is tempting to believe that one can guarantee exactness for the strengthened SDP relaxation for more instances.

6 Conclusions

We have presented new conditions for the exactness of the SDP relaxation of homogeneous QCQPs with forest structure. These can be checked a priori and in polynomial time by solving $n-1$ SOCPs, where n is the number of variables in the problem. When the conditions hold for a given problem, we propose a new way to guarantee exactness for problems that are similar to the given problem. When our conditions do not hold for a given problem that does have an exact SDP relaxation and satisfies a technical condition, we explore a way to guarantee exactness for problems with data $\{\alpha_k\}_{k=1}^m$ that are similar to those in the given problem. We presented a numerical example to demonstrate the theory.

References

[1] Godai Azuma, Mituhiro Fukuda, Sunyoung Kim, and Makoto Yamashita. Exact SDP relaxations of quadratically constrained quadratic programs with forest structures, September 2020. arXiv:2009.02638.

[2] Xiaowei Bao, Nikolaos V. Sahinidis, and Mohit Tawarmalani. Semidefinite relaxations for quadratically constrained quadratic programming: A review and comparisons. *Mathematical Programming*, 129(1):129–157, May 2011.

[3] Alexander I. Barvinok. Problems of distance geometry and convex properties of quadratic maps. *Discrete & Computational Geometry*, 13(2):189–202, 1995.

[4] A. Ben-Tal, L. El Ghaoui, and A.S. Nemirovski. *Robust Optimization*. Princeton Series in Applied Mathematics. Princeton University Press, October 2009.

[5] Subhonmesh Bose, Dennice F. Gayme, K. Mani Chandy, and Steven H. Low. Quadratically Constrained Quadratic Programs on Acyclic Graphs With Application to Power Flow. *IEEE Transactions on Control of Network Systems*, 2(3):278–287, Sep 2015.

[6] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, March 2004.

[7] Samuel Burer. A gentle, geometric introduction to copositive optimization. *Mathematical Programming*, 151(1):89–116, Mar 2015.

[8] Samuel Burer and Yinyu Ye. Exact semidefinite formulations for a class of (random and non-random) nonconvex quadratic programs. *Mathematical Programming*, Feb 2019.

[9] Chen Chen, Alper Atamtürk, and Shmuel S. Oren. A spatial branch-and-cut method for nonconvex QCQP with bounded complex variables. *Mathematical Programming*, 165(2, Ser. A):549–577, 2017.

[10] Diego Cifuentes, Sameer Agarwal, Pablo A. Parrilo, and Rekha R. Thomas. On the local stability of semidefinite relaxations, August 2020. arXiv:1710.04287v3.

[11] Diego Cifuentes, Corey Harris, and Bernd Sturmfels. The geometry of SDP-exactness in quadratic optimization. *Mathematical Programming*, 182(1-2):399–428, May 2019.

[12] Jerome Dancis. Positive semidefinite completions of partial hermitian matrices. *Linear Algebra and its Applications*, 175:97–114, October 1992.

[13] L. Gan, N. Li, U. Topcu, and S. Low. On the exactness of convex relaxation for optimal power flow in tree networks. In *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*, pages 465–471, 2012.

[14] L. Gan, N. Li, U. Topcu, and S. H. Low. Exact Convex Relaxation of Optimal Power Flow in Radial Networks. *IEEE Transactions on Automatic Control*, 60(1):72–87, 2015.

[15] Michel X. Goemans and David P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the ACM*, 42(6):1115–1145, November 1995.

[16] Rujun Jiang and Duan Li. Second order cone constrained convex relaxations for nonconvex quadratically constrained quadratic programming. *Journal of Global Optimization*, 75(2):461–494, June 2019.

[17] Aritra Konar and Nicholas D. Sidiropoulos. Hidden Convexity in QCQP with Toeplitz-Hermitian Quadratics. *IEEE Signal Processing Letters*, 22(10):1623–1627, Oct 2015.

[18] Monique Laurent and Antonios Varvitsiotis. A new graph parameter related to bounded rank positive semidefinite matrix completions. *Mathematical Programming*, 145(1-2):291–325, Feb 2013.

[19] Javad Lavaei, David Tse, and Baosen Zhang. Geometry of power flows and optimization in distribution networks. *IEEE Transactions on Power Systems*, 29(2):572–583, Mar 2014.

[20] Alex Lemon, Anthony Man-Cho So, and Yinyu Ye. Low-rank semidefinite programming: Theory and applications. *Foundations and Trends® in Optimization*, 2(1-2):1–156, 2016.

[21] Z. Luo, W. Ma, A. M. So, Y. Ye, and S. Zhang. Semidefinite relaxation of quadratic optimization problems. *IEEE Signal Processing Magazine*, 27(3):20–34, 2010.

[22] M. Nick, R. Cherkaoui, J. L. Boudec, and M. Paolone. An exact convex formulation of the optimal power flow in radial distribution networks including transverse components. *IEEE Transactions on Automatic Control*, 63(3):682–697, 2018.

[23] Gábor Pataki. On the rank of extreme matrices in semidefinite programs and the multiplicity of optimal eigenvalues. *Mathematics of operations research*, 23(2):339–358, 1998.

[24] N. Z. Shor. Quadratic optimization problems. *Soviet Journal of Computer and Systems Sciences*, 25:1–11, 1987.

[25] Hein van der Holst. Graphs whose positive semi-definite matrices have nullity at most two. *Linear Algebra and its Applications*, 375:1–11, Dec 2003.

[26] Lieven Vandenberghe and Martin S. Andersen. Chordal Graphs and Semidefinite Optimization. *Foundations and Trends® in Optimization*, 1(4):241–433, 2015.

[27] Stephen A. Vavasis. Quadratic programming is in NP. *Information Processing Letters*, 36(2):73–77, October 1990.

[28] Alex L. Wang and Fatma Kilinc-Karzan. On the tightness of SDP relaxations of QCQPs, November 2019. arXiv:1911.09195.

Appendix D

Details for Chapter 3 and Paper C

D.1 Feasibility System as a SOCP

In this section we outline how to solve a feasibility system as a second-order cone program.

Let

$$\mathcal{Q}^n = \left\{(t, x) \in \mathbb{R} \times \mathbb{R}^{n-1} \mid t \geq \|x\|_2\right\}$$

denote the second-order cone of order n and let

$$\mathcal{Q}_r^n = \left\{(s, t, x) \in \mathbb{R} \times \mathbb{R} \times \mathbb{R}^{n-2} \mid 2st \geq \|x\|_2^2\,,\ s, t \geq 0\right\}$$

denote the rotated second-order cone, such that

$$x \in \mathcal{Q}^n \ \Leftrightarrow\ T_n x \in \mathcal{Q}_r^n$$

where

$$T_n = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 & 0 \\ 1 & -1 & 0 \\ 0 & 0 & \sqrt{2} I_{n-2} \end{bmatrix}.$$

Fixing i and j, a single feasibility system takes the form

$$\begin{aligned} &\text{find} && \lambda \\ &\text{subject to} && E_{ij} \bullet Y(\lambda) = 0 \\ &&& \bar{E}_{ij} \bullet Y(\lambda) = 0 \\ &&& Y(\lambda) \succeq 0,\ \lambda \succeq 0. \end{aligned} \tag{D.1}$$

Letting E denote the aggregate sparsity pattern of the matrices $\{A_k\}_{k=0}^m$, and S_E^n denote the set of symmetric matrices of order n with sparsity pattern E, we can formulate the problem as

$$\begin{aligned} &\text{find} && \lambda \\ &\text{subject to} && E_{ij} \bullet Y(\lambda) = 0 \\ &&& \bar{E}_{ij} \bullet Y(\lambda) = 0 \\ &&& Y(\lambda) \in \mathcal{S}_+^n \cap \mathcal{S}_E^n,\ \lambda \succeq 0. \end{aligned} \tag{D.2}$$

Since the sparsity graph of E is a tree, and therefore chordal, the cone of positive semidefinite matrices with sparsity pattern E can be decomposed ([83], Theorem 9.2) as

$$Y(\lambda) = \sum_{\beta \in C} P_\beta^T H_\beta P_\beta,\ H_\beta \succeq 0 \tag{D.3}$$

where C is the set of cliques, and the matrix P_β "picks out" the right elements to be multiplied with the clique. This means that we can replace the PSD constraint on $Y(\lambda)$ with PSD constraints on smaller matrices for each clique, so long as we make sure that the sum of overlapping elements is consistent with the element in $Y(\lambda)$. For our specific problem (homogeneous QCQPs with forest structure) all cliques consist of two nodes, so we can denote the set of cliques by

$$C = \{(k,l) : (k,l) \in \mathcal{E}\},$$

so the decomposition (D.3) consists of $n-1$ 2×2 matrices. Denoting the elements of these matrices by

$$H_{kl} = \begin{bmatrix} \alpha_{kl} & \gamma_{kl} \\ \gamma_{kl} & \beta_{kl} \end{bmatrix},$$

the constraint $H_{kl} \succeq 0$ is equivalent to the rotated quadratic cone constraint (nonnegative determinant)

$$\alpha_{kl}\beta_{kl} \geq |\gamma_{kl}|^2 \Leftrightarrow \begin{bmatrix} \alpha_{kl} \\ \beta_{kl} \\ \sqrt{2}\,\mathrm{Re}(\gamma_{kl}) \\ \sqrt{2}\,\mathrm{Im}(\gamma_{kl}) \end{bmatrix} \in \mathcal{Q}_r^4.$$

In conclusion the constraint $Y \in \mathcal{S}_+^n \cap \mathcal{S}_E^n$ can be replaced by $n-1$ rotated second-order cone constraints, and $2n-1$ equality constraints (n diagonal elements and $n-1$ off-diagonal elements). The off-diagonal elements are non-overlapping, so the sum consists of a single element, becoming

$$Y_{kl} = \gamma_{kl}.$$

The diagonal elements give rise to the equality constraints

$$Y_{kk} = \sum_{l \in \mathrm{adj}^+(k)} \alpha_{kl} + \sum_{l \in \mathrm{adj}^-(k)} \beta_{kl}, \tag{D.4}$$

where $\mathrm{adj}^+(k)$ and $\mathrm{adj}^-(k)$ denotes the upper and lower adjacency sets for vertex k with respect to some ordering. Using these in place of the matrix equality, the feasibility problem becomes the SOC given by

$$\text{find} \quad \lambda \in \mathbb{R}^m \tag{D.5a}$$

$$\text{subject to} \quad Y_{ij} = [A_0]_{ij} + \sum_{p=1}^{m} \lambda_p [A_p]_{ij} = 0 \tag{D.5b}$$

$$Y_{kl} = [A_0]_{kl} + \sum_{p=1}^{m} \lambda_p [A_p]_{kl}, \qquad k = l \ \vee \ (k,l) \in \mathcal{E} \tag{D.5c}$$

$$Y_{kl} = \gamma_{kl}, \qquad (k,l) \in \mathcal{E} \tag{D.5d}$$

$$Y_{kk} = \sum_{l \in \mathrm{adj}^+(k)} \alpha_{kl} + \sum_{l \in \mathrm{adj}^-(k)} \beta_{kl}, \qquad k = 1, 2, \ldots, n \tag{D.5e}$$

$$\begin{bmatrix} \alpha_{kl} \\ \beta_{kl} \\ \sqrt{2}\,\mathrm{Re}(\gamma_{kl}) \\ \sqrt{2}\,\mathrm{Im}(\gamma_{kl}) \end{bmatrix} \in \mathcal{Q}_r^4, \qquad (k,l) \in \mathcal{E} \tag{D.5f}$$

$$\lambda \succeq 0 \tag{D.5g}$$

Note that the equalities (D.5b)–(D.5d) are complex when the matrices are complex.

Appendix E

Details for Chapter 4 and Paper B

E.1 Implementation

Here we outline some details for the implementation of the separation problem of paper B. An implementation can be found at https://github.com/A-Eltved/strengthened_sdr.

E.1.1 Generating Instances with a Known Interior Point

The separation procedure assumes that $\mathcal{F}$ has a non-empty interior. A way to ensure that this is the case for our experiments is to generate instances with a known interior point $\hat{x}$ that can be used in the separation, where we need the rank-1 matrix $\hat{Y}$. The important observation that makes this possible is that we can choose a after everything else is fixed to make sure that $\mathcal{F}$ has interior (e.g., $a \to -\infty$ and $r < R$).

Let $\mathcal{U}(l, u)$ denote the uniform probability distribution over the interval $[l, u]$ and let $\mathcal{N}(\mu, \Sigma)$ denote the normal distribution with mean μ and covariance Σ.

We can generate instances by following Algorithm 1.

Algorithm 1: Generate instances with a known interior point

Result: $r, R, a \in \mathbb{R}$; $b, c \in \mathbb{R}^n$; $\hat{x} \in \text{int}(\mathcal{F})$
Input: Problem dimension n;
Parameters: $\beta = 1$ (width of uniform distribution for a);
$R \leftarrow 1$;
$r \leftarrow \mathcal{U}(0, R)$ (draw from uniform distribution);
$\tilde{x} \leftarrow \mathcal{N}(0, I)$;
$\tilde{r} \leftarrow \mathcal{U}(r, R)$;
$\hat{x} \leftarrow \frac{\tilde{x}}{\|\tilde{x}\|}\tilde{r}$;
$b \leftarrow \mathcal{N}(0, I)$;
$c \leftarrow \mathcal{N}(0, I)$;
$a_{\max} \leftarrow b^T\hat{x} - \|\hat{x} - c\|$;
$a \leftarrow \mathcal{U}(a_{\max} - \beta, a_{\max})$;

E.1.2 Computing $[c]_{\max}$

We compute $[c]_{\max}$ by doing bisection over the interval $[0, \|c\|]$ as outlined in Algorithm 2.

Algorithm 2: Compute $[c]_{\max}$ by bisection

Result: $[c]_{\max}$
Input: problem data $r, R, a \in \mathbb{R}$ and $b, c \in \mathbb{R}^n$;
Parameters: $\epsilon = 10^{-5}$ (tolerance for bisection interval);
Local variables: l, u (lower and upper bound) and μ, v;
$l \leftarrow 0$;
$u \leftarrow \|c\|$;
while $u - l > \epsilon$ **do**
  $\mu \leftarrow l + \frac{u-l}{2}$;
  $v \leftarrow \min\{\mu x^T x - rc^T x : \|x\| \leq R, \|x - c\| \leq b^T x - a\}$;
  if $v < 0$ **then**
    $l \leftarrow \mu$;
  else
    $u \leftarrow \mu$;
  end
end
$[c]_{\max} \leftarrow u$;

E.1.3 The Cone $\widehat{\mathcal{R}}$

For the separation problem we need the cone $\widehat{\mathcal{R}} \subseteq \mathcal{S}_+^{n+1}$ for the chosen relaxation. In the following we state this cone for the Shor relaxation $\mathcal{R}_{\text{shor}}$ and the Shor relaxation intersected with the KSOC constraint $\mathcal{R}_{\text{shor}} \cap \mathcal{R}_{\text{ksoc}}$.

E.1.3.1 Shor

We wish to implement the separation procedure for the relaxation $\mathcal{R}_{\text{shor}}$ of (4.6), which takes the form

$$\begin{aligned}
\min \quad & H \bullet X + 2\, g^T x && \text{(E.1a)}\\
\text{subject to} \quad & \gamma^2 \leq \operatorname{tr}(X) \leq \nu^2 && \text{(E.1b)}\\
& \operatorname{tr}(X) - 2\, c^T x + c^T c \leq bb^T \bullet X - 2\,\alpha\, b^T x + \alpha^2 && \text{(E.1c)}\\
& 0 \leq b^T x - \alpha && \text{(E.1d)}\\
& Y(x, X) \succeq 0 && \text{(E.1e)}
\end{aligned}$$

To do the separation we need to write it in the form of $Y(x, X) \in \widehat{\mathcal{R}}$, where $\widehat{\mathcal{R}}$ is a closed convex cone. Define the matrices

$$\begin{aligned}
A_1 &:= \begin{pmatrix} -\gamma^2 & 0 \\ 0 & I \end{pmatrix}\\
A_2 &:= \begin{pmatrix} \nu^2 & 0 \\ 0 & -I \end{pmatrix}\\
A_3 &:= \begin{pmatrix} \alpha^2 - c^T c & c^T - \alpha b^T \\ c - \alpha b & bb^T - I \end{pmatrix}\\
A_4 &:= \begin{pmatrix} -\alpha & \frac{1}{2} b^T \\ \frac{1}{2} b & 0 \end{pmatrix}
\end{aligned}$$

and the half spaces

$$\begin{aligned}
H_1 &:= \{Y(x, X) : A_1 \bullet Y(x, X) \geq 0\}\\
H_2 &:= \{Y(x, X) : A_2 \bullet Y(x, X) \geq 0\}\\
H_3 &:= \{Y(x, X) : A_3 \bullet Y(x, X) \geq 0\}\\
H_4 &:= \{Y(x, X) : A_4 \bullet Y(x, X) \geq 0\}.
\end{aligned}$$

Let $\widehat{\mathcal{R}}_{\text{shor}} := \mathcal{S}_+^{n+1} \cap H_1 \cap H_2 \cap H_3 \cap H_4$, where $\mathcal{S}_+^{n+1}$ denotes the cone of symmetric positive semidefinite matrices of order $n + 1$, and note that this is the intersection of closed cones and therefore closed. Then we can write the

Shor relaxation as

$$\min \quad \begin{pmatrix} 0 & g^T \\ g & H \end{pmatrix} \bullet Y(x,X) \tag{E.2a}$$

$$\text{subject to} \quad Y(x,X) \in \widehat{\mathcal{R}}_{\text{shor}} \tag{E.2b}$$

Using that, for a pair of cones K_1 and K_2, the dual of the intersections is $(K_1 \cap K_2)^* = K_1^* + K_2^*$ (where $+$ denotes set addition $\{u+v : u \in K_1^*, v \in K_2^*\}$), we have

$$\begin{aligned} \widehat{\mathcal{R}}_{\text{shor}}^* &= (\mathcal{S}_+^{n+1} \cap H_1 \cap H_2 \cap H_3 \cap H_4)^* \\ &= (\mathcal{S}_+^{n+1})^* + H_1^* + H_2^* + H_3^* + H_4^* \\ &= \mathcal{S}_+^{n+1} + R_1 + R_2 + R_3 + R_4 \end{aligned}$$

where R_i is the ray given by

$$R_i := \{\zeta A_i : \zeta \geq 0\}$$

For the Shor relaxation, we have the separation problem:

$$\begin{aligned} \min \quad & C_q \bullet J_q + C_l \bullet J_l - \operatorname{tr}(\bar{X})[q+l]_{\min} \\ \text{subject to} \quad & J_q \in \widehat{\mathcal{R}}^*, \quad J_l \in \widehat{\mathcal{R}}^*, \quad J_q + J_l - \begin{pmatrix} [q+l]_{\min} & 0^T \\ 0 & 0 \end{pmatrix} \in \widehat{\mathcal{R}}^* \\ & [J_l]_{2:n+1,2:n+1} = 0, \quad [q+l]_{\min} \geq 0, \quad \hat{Y} \bullet J_q \leq 1, \quad \hat{Y} \bullet J_l \leq 1 \end{aligned}$$

where the variables are $J_l, J_q \in \mathcal{S}^{n+1}$ and $[q+l]_{\min} \in \mathbb{R}$. The data is

$$C_q := R^2 \begin{pmatrix} 1 & \bar{x}^T \\ \bar{x} & \bar{X} \end{pmatrix}$$

and

$$C_l := \begin{pmatrix} [C_l]_{11} & [C_l]_{\bullet 1}^T \\ [C_l]_{\bullet 1} & 0 \end{pmatrix}.$$

where

$$\begin{aligned} [C_l]_{11} &:= (r+R)(b^T\bar{x} - a) - rR + c^T\bar{x} + [c]_{\max}R, \\ [C_l]_{\bullet 1} &:= (r+R)(\bar{X}b - a\bar{x}) - rR\bar{x} + \bar{X}c + [c]_{\max}R\bar{x}. \end{aligned}$$

For the $r = 0$ cut mentioned in Corollary 1 of Paper B, the objective is different in the separation: we have

$$C_l := \begin{pmatrix} R(b^T\bar{x} - a) + c^T\bar{x} + [c]_{\max}R & \left(R(\bar{X}b - a\bar{x}) + \bar{X}c + [c]_{\max}R\bar{x}\right)^T \\ R(\bar{X}b - a\bar{x}) + \bar{X}c + [c]_{\max}R\bar{x} & 0 \end{pmatrix},$$

while C_q stays the same.

With the Shor dual cone given, we can write the separation problem as

$$
\begin{aligned}
\min \quad & C_q \bullet J_q + C_l \bullet J_l - \operatorname{tr}(\bar{X})[q+l]_{\min} \\
& J_q = Z_q + \sum_{i=1}^{4} \zeta_q^i A_i \\
& Z_q \succeq 0,\ \zeta_q^i \geq 0,\ i = 1, \ldots, 4 \\
& J_l = Z_l + \sum_{i=1}^{4} \zeta_l^i A_i \\
& Z_l \succeq 0,\ \zeta_l^i \geq 0,\ i = 1, \ldots, 4 \\
& J_q + J_l - \begin{pmatrix} [q+l]_{\min} & 0^T \\ 0 & 0 \end{pmatrix} = Z_{q+l} + \sum_{i=1}^{4} \zeta_{q+l}^i A_i \\
& Z_{q+l} \succeq 0,\ \zeta_{q+l}^i \geq 0,\ i = 1, \ldots, 4 \\
& [J_l]_{2:n+1,2:n+1} = 0, \quad [q+l]_{\min} \geq 0, \quad \hat{Y} \bullet J_q \leq 1, \quad \hat{Y} \bullet J_l \leq 1
\end{aligned}
$$

where the variables are $J_l, J_q, Z_q, Z_l, Z_{q+l} \in \mathcal{S}^{n+1}$ and $[q+l]_{\min}, \zeta_q^1, \ldots, \zeta_q^4, \zeta_l^1, \ldots, \zeta_l^4, \zeta_{q+l}^1, \ldots, \zeta_{q+l}^4, \in \mathbb{R}$.

E.1.3.2 Shor and KSOC

We wish to implement the separation procedure for the relaxation $\mathcal{R}_{\text{shor}} \cap \mathcal{R}_{\text{ksoc}}$ of (4.6), which takes the form

$$
\begin{aligned}
\min \quad & H \bullet X + 2\, g^T x && \text{(E.3a)} \\
\text{subject to} \quad & r^2 \leq \operatorname{tr}(X) \leq R^2 && \text{(E.3b)} \\
& \operatorname{tr}(X) - 2\, c^T x + c^T c \leq b b^T \bullet X - 2\, a\, b^T x + a^2 && \text{(E.3c)} \\
& 0 \leq b^T x - a && \text{(E.3d)} \\
& Y(x, X) \succeq 0 && \text{(E.3e)} \\
& \begin{pmatrix} R & x^T \\ x & R\, I \end{pmatrix} \otimes \begin{pmatrix} b^T x - a & x^T - c^T \\ x - c & (b^T x - a) I \end{pmatrix} \succeq 0. && \text{(E.3f)}
\end{aligned}
$$

To do the separation we need to write it in the form of $Y(x, X) \in \widehat{\mathcal{R}}$, where $\widehat{\mathcal{R}}$ is a closed convex cone.

Compared to the Shor cone in the previous section we need to incorporate the Kronecker constraint (E.3f). To this end, we define a $(n+1)^2 \times (n+1)^2$ matrix,

B, where each element is formed as the inner product of a matrix B_{ijpq} and $Y(x, X)$. The structure of B is block-arrow with arrow $n+1 \times n+1$ blocks and the pq index determines the block while the ij index determines the element within that block. We will use zero-indexing such that the indices run from $0, 1, \ldots, n$.

The diagonal elements of the large matrix are formed with the matrix

$$B_{ijpq} = \frac{R}{2}\begin{pmatrix} -2a & b^T \\ b & 0 \end{pmatrix}, i = j, p = q.$$

The row and column elements of this diagonal block is given by

$$B_{ijpq} = B_{jipq} = \frac{1}{2}\begin{pmatrix} 0 & -ae_j^T \\ -ae_j^T & e_j b^T + be_j^T \end{pmatrix}, i = 0, j = 1, \ldots, n, p = q.$$

The qth block row (column) of the first block column (row) has the diagonal

$$B_{ijpq} = B_{ijpq} = \frac{R}{2}\begin{pmatrix} -2c_q & e_q^T \\ e_q & 0 \end{pmatrix}, \; i = j, p = 0, q > 0.$$

The first column/row of these blocks are given by

$$B_{ijpq} = B_{ijpq} = \frac{1}{2}\begin{pmatrix} 0 & -c_q e_j^T \\ -c_q e_j & e_j e_q^T + e_q e_j^T \end{pmatrix}, \; i = 0, j > 0, p = 0, q > 0.$$

For all other elements we have

$$B_{ijpq} = 0.$$

We can now form the elements of the large matrix with elements

$$B_{p(n+1)+i,q(n+1)+j} = B_{ijpq} \bullet Y(x, X).$$

We will denote the matrix as $(B_{ijpq} \bullet Y(x, X))$ to emphasize the dependency on $Y(x, X)$. With this we can define the cone

$$K_{\text{ksoc}} := \{Y(x, X) : (B_{ijpq} \bullet Y(x, X)) \succeq 0\}.$$

Then we can define the closed convex cone

$$\widehat{\mathcal{R}}_{\text{shor+ksoc}} := \mathcal{S}_+^{n+1} \cap H_1 \cap H_2 \cap H_3 \cap H_4 \cap K_{\text{ksoc}}$$

with dual cone

$$\widehat{\mathcal{R}}^*_{\text{shor+ksoc}} := \mathcal{S}_+^{n+1} + R_1 + R_2 + R_3 + R_4 + K^*_{\text{ksoc}},$$

where

$$K^*_{\text{ksoc}} = \{\sum_{ijpq} w_{ijpq} B_{ijpq} : W = (w_{ijpq}) \succeq 0\}.$$

Here the sum runs over the indices of the large matrix, so that it has $(n+1)^2(n+1)^2 = (n+1)^4$ terms.

With the dual cone given, we can write the separation problem as

$$\begin{aligned}
\min \quad & C^q \bullet J^q + C^l \bullet J^l - \operatorname{tr}(\bar{X})[q+l]_{\min} \\
& J^q = Z^q + \sum_{i=1}^{4} \zeta_i^q A_i + \sum_{ijpq} w_{ijpq}^q B_{ijpq} \\
& Z^q \succeq 0,\ \zeta_i^q \geq 0,\ i = 1, \ldots, 4,\ (w_{ijpq}^q) \succeq 0 \\
& J^l = Z^{l\mathcal{S}_+} + \sum_{i=1}^{4} \zeta_i^l A_i + \sum_{ijpq} w_{ijpq}^l B_{ijpq} \\
& Z^l \succeq 0,\ \zeta_i^l \geq 0,\ i = 1, \ldots, 4,\ (w_{ijpq}^l) \succeq 0 \\
& J^q + J^l - \begin{pmatrix} [q+l]_{\min} & 0^T \\ 0 & 0 \end{pmatrix} = Z^{q+l} + \sum_{i=1}^{4} \zeta_i^{q+l} A_i + \sum_{ijpq} w_{ijpq}^{q+l} B_{ijpq} \\
& Z^{q+l} \succeq 0,\ \zeta_i^{q+l} \geq 0,\ i = 1, \ldots, 4,\ (w_{ijpq}^{q+l}) \succeq 0 \\
& H^l = 0, \quad [q+l]_{\min} \geq 0, \quad \hat{Y} \bullet J^q \leq 1, \quad \hat{Y} \bullet J^l \leq 1
\end{aligned}$$

E.2 Separation of Slab Inequalities in the Convex Case

In this section we consider the special case when $\gamma = 0$ and restrict the nonnegative functions to be given by a slab ($q(x) := \mu - s^T x$ and $l(x) := s^T x - \lambda$) as described in Section 2.1 of Paper B. Note that we can take $[c]_{\max} = 0$, so the cuts become

$$R^2(\mu - s^T x) + R(bs^T \bullet X - as^T x - \lambda(b^T x - a)) - (\mu - \lambda)\operatorname{tr}(X) + cs^T \bullet X - \lambda c^T x \geq 0. \tag{E.4}$$

The following Theorem and proof describes the separation procedure in this special case.

Theorem E.1 *If $r = 0$, the slab inequalities (E.4) are separable in polynomial time.*

PROOF. We will refer to a slab by its tuple (λ, s, μ). The proof is motivated by the observation that λ, s, and μ appear linearly in (E.4). Let $(\bar{x}, \bar{X})$ be given. We wish to determine whether $(\bar{x}, \bar{X}) \in \mathcal{R}_{\text{slab}}$ and, if not, to find a slab (λ, s, μ) and corresponding inequality (E.4) separating $(\bar{x}, \bar{X})$ from $\mathcal{R}_{\text{slab}}$.

Let $f_{\bar{x},\bar{X}}(\lambda, s, \mu)$ denote the linear function of (λ, s, μ) defining the left-hand side of (E.4) when (x, X) is fixed at the values $(\bar{x}, \bar{X})$, and consider the optimization

$$\underset{\lambda,s,\mu}{\text{minimize}}\ f_{\bar{x},\bar{X}}(\lambda, s, \mu) \tag{E.5a}$$

$$\text{subject to}\ \lambda \leq \min_{x \in \mathcal{F}} s^T x \tag{E.5b}$$

$$\max_{x \in \mathcal{F}} s^T x \leq \mu \tag{E.5c}$$

Note that (E.5b)–(E.5c) enforce that (λ, s, μ) is a valid slab. If the optimal value of (E.5) is negative, then any optimal slab (λ, s, μ) yields a SI (E.4) that cuts off $(\bar{x}, \bar{X})$. On the other hand, if the optimal objective value is nonnegative, then we have proven $(\bar{x}, \bar{X}) \in \mathcal{R}_{\text{slab}}$.

We claim that (E.5) can be expressed as a SOCP of polynomial size. First, as mentioned above, the objective is linear. Second, consider the constraint (E.5b), which ensures that λ is no larger than the minimum value of $s^T x$ over $x \in \mathcal{F}$. With both s and x varying, (E.5b) contains the bilinear term $s^T x$. However, as is standard, this constraint is equivalent to forcing λ to be no larger than the objective value of a feasible point for the dual of $\min\{s^T x : x \in \mathcal{F}\}$:

$$\underset{\lambda_1,\lambda_2 \in \mathbb{R},\ y^1,y^2 \in \mathbb{R}^n}{\text{maximize}}\ -R\lambda_1 + a\lambda_2 + c^T y^2 \tag{E.6a}$$

$$\text{subject to}\ s = \lambda_2 b + y^1 + y^2 \tag{E.6b}$$

$$\|y^1\| \leq \lambda_1,\ \|y^2\| \leq \lambda_2 \tag{E.6c}$$

Constraint (E.5c) can be handled similarly. The resulting SOCP is

$$\underset{\lambda,s,\mu,\lambda_1,\lambda_2,y^1,y^2,\mu_1,\mu_2,z^1,z^2}{\text{minimize}}\ f_{\bar{x},\bar{X}}(\lambda, s, \mu) \tag{E.7a}$$

$$\text{subject to}\ \lambda \leq -R\lambda_1 + a\lambda_2 + c^T y^2 \tag{E.7b}$$

$$s = \lambda_2 b + y^1 + y^2 \tag{E.7c}$$

$$\|y^1\| \leq \lambda_1,\ \|y^2\| \leq \lambda_2 \tag{E.7d}$$

$$\mu \geq R\mu_1 - a\mu_2 - c^T z^2 \tag{E.7e}$$

$$s = -\mu_2 b - z^1 - z^2 \tag{E.7f}$$

$$\|z^1\| \leq \mu_1,\ \|z^2\| \leq \mu_2 \tag{E.7g}$$

We can solve this SOCP in polynomial time, which means that we can separate the slab inequalities over all valid slabs in polynomial time.

E.3 Orthogonal Inequalities

In this section we present some results for the orthogonal generalization described in Section 4.2 for a special case of problem (4.6) where $r = 0$. In this case, the SOCs can be combined with an orthogonal matrix $Q \in \mathcal{O}_n$ to form

$$b - a^T x + Q \bullet (x - c)x^T = \begin{pmatrix} 1 \\ Qx \end{pmatrix}^T \begin{pmatrix} b^T x - a \\ x - c \end{pmatrix} \geq 0.$$

Relaxing, we obtain the valid inequality

$$b^T x - a + Q \bullet (X - cx^T) \geq 0. \tag{E.8}$$

We call the class of all such inequalities the *orthogonal inequalities* (abbreviated *OIs*), and we define $\mathcal{R}_{\text{orth}}$ to be the set of all (x, X) satisfying all OIs.

The following result establishes that the OIs are separable in polynomial time.

Theorem E.2 *The orthogonal inequalities (E.8) are separable in polynomial time over all $Q \in \mathcal{O}_n$ at the cost of a singular value decomposition of size $n \times n$.*

Proof. With (x, X) at fixed values $(\bar{x}, \bar{X})$, the separation problem involves minimizing the linear objective $b^T \bar{x} - \alpha + Q \bullet (\bar{X} - c\bar{x}^T)$ over $Q \in \mathcal{O}_n$. As $\mathcal{O}_n$ is a compact set, let Q^* be an optimal orthogonal matrix. If the optimal value at Q^* is negative, then we have discovered a violated cut using Q^*; otherwise, we have proven that $(\bar{x}, \bar{X}) \in \mathcal{R}_{\text{orth}}$. Hence, separation amounts to linear minimization over $\mathcal{O}_n$.

We next argue that, for any square matrix $M \in \mathbb{R}^{n \times n}$, $M \bullet Q$ can be minimized over $Q \in \mathcal{O}_n$ in polynomial time. First, suppose that M is nonnegative diagonal so that $M \bullet Q = M_{11}Q_{11} + \cdots + M_{nn}Q_{nn}$ with every $M_{jj} \geq 0$. Noting that each Q_{jj} satisfies $|Q_{jj}| \leq 1$, then Q^* should be diagonal with each $Q^*_{jj} = -1$, i.e., $Q^* = -I$, to ensure that $M \bullet Q^*$ is indeed minimum. Now suppose that M is not diagonal, and let $M = U\Sigma V^T$ be its singular value decomposition with U, V orthogonal and Σ nonnegative diagonal. Then the change of variables $\tilde{Q} = VQU^T$ shows that $\min\{M \bullet Q : Q \in \mathcal{O}_n\}$ is equivalent to $\min\{\Sigma \bullet \tilde{Q} : \tilde{Q} \in \mathcal{O}_n\}$, i.e., the non-diagonal case reduces to the diagonal case.

The following result establishes that the orthogonal generalization of the slab inequalities (E.4) does not improve the relaxation with just the slab inequalities when $c = 0$.

Proposition E.3 *When $c = 0$, the optimal orthogonal matrix is $Q^* = -I$.*

PROOF. Suppose $c = 0$. Then the orthogonal generalization of the slab inequalities (E.4) is

$$R^2(\mu - s^T x) + Rsb^T \bullet X - Ras^T x - R\lambda(b^T x - a) + Q \bullet ((\mu - \lambda)X) \geq 0. \quad \text{(E.9)}$$

Minimizing this over $Q \in \mathcal{O}_n$ for fixed (x, X) corresponds to minimizing the term $Q \bullet X$, since $\mu - \lambda \geq 0$. Since $X \succeq 0$, we can make an eigenvalue decomposition as $X = V\Sigma V^T$, where V is an orthogonal matrix and Σ is diagonal with nonnegative diagonal elements. Hence, we have

$$Q \bullet X = \operatorname{tr}(Q^T X) = \operatorname{tr}(Q^T V \Sigma V^T) = \operatorname{tr}(V^T Q^T V \Sigma).$$

Since Q and V are both orthogonal and Σ is diagonal with nonnegative elements, we must have

$$V^T Q^T V = -I \iff Q = -I,$$

as desired.

Appendix F

Details for Chapter 5

F.1 Upper Bound on Squared Current Magnitude

One way to obtain an upper bound is to consider the power injection and the voltage bounds at the node. From the power balance (5.1b) we have that

$$i_k^* = \frac{\tilde{s}_k}{v_k} \tag{F.1}$$

Lemma 1 *For two complex numbers $a, b \in \mathbb{C}$ we have*

$$\left|\frac{a}{b}\right| = \left|\frac{|a|\, e^{i\theta_a}}{|b|\, e^{i\theta_b}}\right| = \frac{|a|}{|b|}\left|\frac{e^{i\theta_a}}{e^{i\theta_b}}\right| = \frac{|a|}{|b|}\left|e^{i(\theta_a-\theta_b)}\right| = \frac{|a|}{|b|}$$

Taking the squared magnitude we have

$$|i_k|^2 = \left|\frac{\tilde{s}_k}{v_k}\right|^2 = \frac{|\tilde{s}_k|^2}{|v_k|^2} \leq \frac{\left|\overline{\tilde{s}_k}\right|^2}{\underline{V}_k^2} \tag{F.2}$$

The upper bound on the complex power injection can be obtained by the generation limits

$$\left|\overline{\tilde{s}_k}\right|^2 = \left(\max\left\{\sum_{g\in G_k}\overline{P}_g - \mathrm{Re}(S_k^d), -\sum_{g\in G_k}\underline{P}_g - \mathrm{Re}(S_k^d)\right\}\right)^2 \tag{F.3}$$

$$+\left(\max\left\{\sum_{g\in G_k}\overline{Q}_g - \mathrm{Im}(S_k^d), -\sum_{g\in G_k}\underline{Q}_g - \mathrm{Im}(S_k^d)\right\}\right)^2 \tag{F.4}$$

F.2 Diagonalization

Here, we outline a diagonalization approach for the OPF problem (5.1). The idea is to exploit the rank-1 nature of $Y^H e_k e_k^T$.

Consider a complex matrix $M = ab^H$, where

$$a = Y^H e_k, b = e_k. \tag{F.5}$$

The matrix can be decomposed into two Hermitian matrices

$$M_+ = \frac{1}{2}\left(ab^H + ba^H\right) \tag{F.6}$$

and

$$M_- = \frac{\jmath}{2}\left(ab^H - ba^H\right). \tag{F.7}$$

The original matrix is given by $M = M_+ + \jmath M_-$. The first matrix is diagonalized by

$$\frac{1}{4}M_+ = (a+b)(a+b)^H - (a-b)(a-b)^H$$

The same vectors ($(a+b)$ and $(a-b)$) can be used to factor $M_- = 2\left(ab^H - ba^H\right)$ as

$$\frac{1}{4}M_- = (a+b)(a-b)^H - (a-b)(a+b)^H$$

The constraint for active/reactive power in these variables is can be expressed as

$$v^H M_+ v = \tilde{p} \tag{F.8}$$

$$v^H M_- v = \tilde{q} \tag{F.9}$$

$$\tag{F.10}$$

Introducing the linear constraints

$$x_{k,1} = (a+b)^H v \tag{F.11}$$

$$x_{k,2} = (a-b)^H v \tag{F.12}$$

we can write (F.8) as

$$x_{k,1}^* x_{k,1} - x_{k,2}^* x_{k,2} = 4\tilde{p}_k \tag{F.13}$$

and (F.9) as

$$x_{k,1}^* x_{k,2} - x_{k,2}^* x_{k,1} = 4\tilde{q}_k \tag{F.14}$$

Note that we introduce two new variables for each bus.