5-2024

Applications of Conic Programming Reformulations

Sarah Kelly
skelly5@clemson.edu

Applications of Conic Programming Reformulations

A Dissertation
Presented to
the Graduate School of
Clemson University

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy
Mathematical Sciences

by
Sarah M. Kelly
May 2024

Accepted by:
Dr. Boshi Yang, Committee Chair
Dr. Hao Hu, Committee Co-Chair
Dr. Yuyuan Ouyang
Dr. Cheng Guo

Abstract

In general, convex programs have nicer properties than nonconvex programs. Notably, in a convex program, every locally optimal solution is also globally optimal. For this reason, there is interest in finding convex reformulations of nonconvex programs. These reformulation often come in the form of a conic program. For example, nonconvex quadratically-constrained quadratic programs (QCQPs) are often relaxed to semidefinite programs (SDPs) and then tightened with valid inequalities. This dissertation gives a few different problems of interest and shows how conic reformulations can be usefully applied.

In one chapter, we consider two variants of the trust-region subproblem. Both variants minimize a quadratic objective, and one variant has a feasible region defined by the intersection of two balls while the other variant has a feasible region defined by the intersection of a ball and a special second-order conic representable set. For each of these variants, we show that their feasible regions can be written as the union of two sub-regions with known lifted convex hulls. We are then able to use this fact to reformulate each variant into a semidefinite program.

In the next chapter, we consider the unit commitment problem as there is much interest in pricing schemes for energy markets with non-convexities. While the mixed-binary programming formulation of the unit commitment problem is not difficult to solve, a convex reformulation is still desired as its shadow prices may be used to formulate a better pricing scheme. From former works, we know that the unit commitment problem can be reformulated as a completely positive program. While this is useful theoretically, solving CPPs is currently still quite challenging. For this reason, we seek to relax the CPP to an

SDP and use its shadow prices in a pricing scheme.

The last application that we look at focuses on identifying implicit inequalities in a linear system of inequalities. While there are several methods that we examine, one of the most efficient methods works by solving a linear program that has been constructed to have a conic feasible region. Such methods may be useful in the preprocessing step of solvers as implicit equalities relate to facial reduction, which helps guarantee a constraint qualification holds.

Table of Contents

Title Page . . . . **i**

Abstract . . . . **ii**

List of Tables . . . . **vi**

List of Figures . . . . **vii**

1 Introduction **1**
1.1 Background on conic programming 2
1.2 Duality 3
1.3 Outline 4

2 Semidefinite Representable Reformulations for Two Variants of the Trust-Region Subproblem **5**
2.1 Introduction 5
2.2 The lifted convex hull 9
2.3 Semidefinite reformulations 11
2.4 Separation and computational results 17

3 Semidefinite Programming Relaxation for Copositive Dual Pricing . . . **23**
3.1 Introduction 23
3.2 CPP and SDP Reformulations of UC 25
3.3 Pricing with SDP Relaxations 29
3.4 Numerical Experiments 35
3.5 Conclusion 38

4 Affine Hull Algorithms **39**
4.1 Introduction 39
4.2 Revisit of the existing methods 41
4.3 Modifications to existing methods 53
4.4 Numerical experiments 59
4.5 Conclusion 67

5 Conclusions and Discussion **69**

Appendices **71**

A Simple examples pertaining to SDP reformulations 72

Bibliography 74

List of Tables

2.1 Cut effectiveness for the $\mathcal{R}_{\text{shor}}$ relaxation of (2.3). 21
2.2 Cut effectiveness for the $\mathcal{R}_{\text{shor}} \cap \mathcal{R}_{\text{ksoc}}$ relaxation of (2.3). 21
2.3 Cut effectiveness for the $\mathcal{R}_{\text{shor}}$ relaxation of (2.2). 22
2.4 Cut effectiveness for the $\mathcal{R}_{\text{shor}} \cap \mathcal{R}_{\text{ksoc}}$ relaxation of (2.2). 22

3.1 UC Method Comparison 36

4.1 Implicit Equalities - Benchmark Results 61
4.2 Implicit Equalities - Random Results 61

List of Figures

1.1 Conic Programming Relations 1

3.1 Load scenarios for IEEE 14-bus system 37
3.2 Line rating scenarios for IEEE 14-bus system 37

Chapter 1

Introduction

This dissertation focuses on conic programming reformulations and their applications. The hierarchy of different types of conic programs is illustrated in Figure 1.1. While linear programs (LPs), second-order cone programs (SOCPs), and semidefinite programs (SDPs) can be solved by interior point methods, not all conic programs can be solved efficiently. Copositive programs are generally NP-hard, and even large-scale SDPs are challenging.

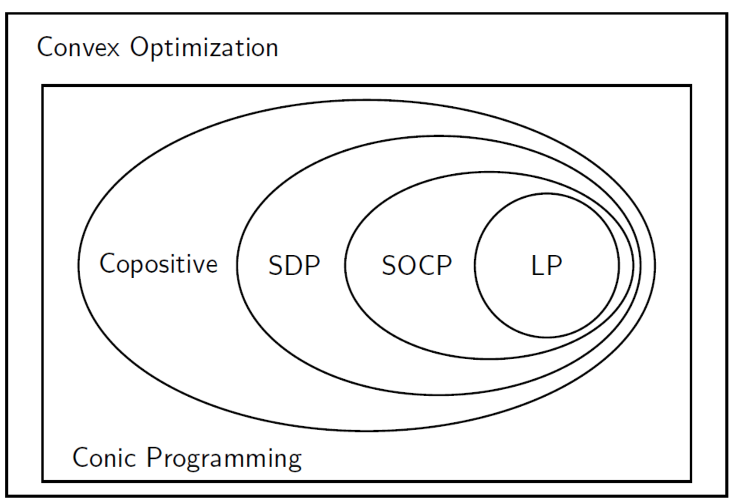


Figure 1.1: Conic Programming Relations

1.1 Background on conic programming

Before giving the general form of a conic program, we first give some relevant definitions:

- A set $\mathcal{K}$ is a *cone* if for any nonnegative scalar α and any $\mathbf{x} \in \mathcal{K}$, $\alpha\mathbf{x} \in \mathcal{K}$.
- A cone $\mathcal{K}$ is *convex* if $\mathbf{x}, \mathbf{y} \in \mathcal{K}$ and $\lambda \in [0, 1]$ implies $\lambda\mathbf{x} + (1 - \lambda)\mathbf{y} \in \mathcal{K}$.
- A cone $\mathcal{K}$ is *pointed* if $-\mathbf{x}, \mathbf{x} \in \mathcal{K}$ implies $\mathbf{x} = \mathbf{0}$.
- A *regular* cone is a cone that is closed, pointed, and convex with a nonempty interior.
- The *dual cone* of a convex cone $\mathcal{K}$ is given by $\mathcal{K}^* := \{\mathbf{y} \mid \langle \mathbf{x}, \mathbf{y} \rangle \geq 0 \text{ for all } \mathbf{x} \in \mathcal{K}\}$.
- A cone $\mathcal{K}$ is a *self-dual* if $\mathcal{K}^* = \mathcal{K}$.

We now give a general form that a conic program takes:

$$
\begin{aligned}
p^* = \min \quad & \langle \mathbf{c}, \mathbf{x} \rangle \\
\text{s.t.} \quad & \langle \mathbf{a}_i, \mathbf{x} \rangle = b_i, \quad i = 1, \ldots, m \\
& \mathbf{x} \succeq_{\mathcal{K}} \mathbf{0}
\end{aligned}
\tag{P}
$$

where $\langle \cdot, \cdot \rangle$ denotes an inner product, $\mathbf{x} \succeq_{\mathcal{K}} \mathbf{y}$ if and only if $\mathbf{x} - \mathbf{y} \in \mathcal{K}$, and $\mathcal{K}$ is a regular cone. Some examples of regular cones include:

- Nonnegative orthant: $\mathbb{R}^n_+ = \{\mathbf{x} \in \mathbb{R}^n \mid \mathbf{x} \geq \mathbf{0}\}$
- Second-order cone: $\mathcal{L}^n = \{\mathbf{x} \in \mathbb{R}^n \mid x_n \geq \sqrt{x_1^2 + \cdots + x_{n-1}^2}\}$
- Positive semidefinite cone: $S^n_+ = \{X \in S^n \mid \mathbf{v}^T X \mathbf{v} \geq 0 \text{ for all } \mathbf{v} \in \mathbb{R}^n\}$
- Copositive cone: $\mathcal{COP}^n = \{X \in S^n \mid \mathbf{v}^T X \mathbf{v} \geq 0 \text{ for all } \mathbf{v} \in \mathbb{R}^n_+\}$
- Completely positive cone: $\mathcal{CP}^n = \{X \in S^n \mid X = BB^T \text{ where } B \geq 0\}$

$\mathbb{R}^n_+, \mathcal{L}^n$, and S^n_+ are self-duals. $\mathcal{COP}^n$ and $\mathcal{CP}^n$ are duals of each other; i.e., $(\mathcal{COP}^n)^* = \mathcal{CP}^n$ and $(\mathcal{CP}^n)^* = \mathcal{COP}^n$.

1.2 Duality

In conic programming, just as in linear programming, there is the concept of duality. The dual problem of (P) can be derived from the viewpoint of Lagrangian duality. For the constraint $b_i - \langle \mathbf{a}_i, \mathbf{x} \rangle = 0$, we associate it with the Lagrangian multiplier y_i. For the constraint $-\mathbf{x} \preceq_{\mathcal{K}} \mathbf{0}$, we associate it with the Lagrangian multiplier $\mathbf{z} \succeq_{\mathcal{K}^*} \mathbf{0}$. The Lagrangian function is

$$\begin{aligned} L(\mathbf{x}; \mathbf{y}, \mathbf{x}) :&= \langle \mathbf{c}, \mathbf{x} \rangle + \sum_{i=1}^{m} y_i (b_i - \langle \mathbf{a}_i, \mathbf{x} \rangle) + \langle \mathbf{z}, -\mathbf{x} \rangle \\ &= \sum_{i=1}^{m} y_i b_i + \langle \mathbf{c} - \sum_{i=1}^{m} y_i \mathbf{a}_i - \mathbf{z}, \mathbf{x} \rangle \end{aligned}$$

The Lagrangian dual function is

$$g(\mathbf{y}, \mathbf{z}) := \min_{\mathbf{x}} L(\mathbf{x}; \mathbf{y}, \mathbf{x}) = \begin{cases} \langle \mathbf{b}, \mathbf{y} \rangle, & \text{if } \mathbf{c} - \sum_{i=1}^{m} y_i \mathbf{a}_i - \mathbf{z} = \mathbf{0} \\ -\infty, & \text{otherwise.} \end{cases}$$

Thus, the dual problem is given by:

$$\max\{g(\mathbf{y}, \mathbf{z}) \mid \mathbf{z} \succeq_{\mathcal{K}^*} \mathbf{0}\},$$

which simplifies to:

$$\begin{aligned} d^* = \max \quad & \langle \mathbf{b}, \mathbf{y} \rangle \\ \text{s.t.} \quad & \mathbf{c} - \sum_{i=1}^{m} y_i \mathbf{a}_i \succeq_{\mathcal{K}^*} \mathbf{0}, \end{aligned} \tag{D}$$

Also as in linear programming, the Weak Duality Theorem and Strong Duality Theorem help us to understand the relationship between a pair of primal-dual conic programs.

Theorem 1 (Weak Duality)**.** *For any* $\mathbf{x}$ *feasible to* (P) *and any* $\mathbf{y}$ *feasible to* (D), $\langle \mathbf{c}, \mathbf{x} \rangle \geq \langle \mathbf{b}, \mathbf{y} \rangle$.

Theorem 2 (Strong Duality)**.** *If* (P) *is bounded below and strictly feasible, then* (D) *is solvable and the optimal values in the problems are equal; i.e.,* $p^* = d^*$*. If* (D) *is bounded above and strictly feasible, then* (P) *is solvable and* $p^* = d^*$*.*

Moreover, from the perspective of sensitivity analysis, we know that the optimal dual solution $\mathbf{y}^*$ can be useful in pricing schemes as y_i^* can be interpreted as the *marginal cost* (or *shadow price*) per one unit increase in b_i.

1.3 Outline

In Chapter 2, we focus on two variants of the trust-region subproblem. These problems that we consider have a convex feasible region, but a quadratic objective function that may be nonconvex. By rewriting each feasible region as a union of two sub-regions with known lifted convex hulls, we are able to obtain exact semidefinite reformulations of these problems.

In Chapter 3, we consider the unit commitment (UC) problem, a nonconvex problem that is used in electricity markets. Utilizing the fact that the UC problem can be reformulated as a completely positive program, we seek to relax the UC problem to a semidefinite program. We do this with the goal of using its shadow prices in a pricing scheme.

In Chapter 4, we look at several different methods for finding implicit equalities in the linear system defining a polyhedron. The ability to efficiently identify implicit equalities is of interest since such methods can be used in facial reduction algorithms, and facial reduction is often a preproceesing step of solvers. As we will see, the methods that perform the best among those discussed in Chapter 4 are the ones based on a linear program with a conic feasible region.

Chapter 2

Semidefinite Representable Reformulations for Two Variants of the Trust-Region Subproblem

Motivated by encouraging numerical results in [21], in this chapter we consider two specific variants of the trust-region subproblem and provide exact semidefinite representable reformulations. The first is over the intersection of two balls; the second is over the intersection of a ball and a special second-order conic representable set. Different from the technique developed in [21], the reformulations in this chapter are based on partitions of the feasible regions into sub-regions with known lifted convex hulls. (This chapter has been published in OR Letters [39].)

2.1 Introduction

The trust-region subproblem (TRS) is a classic nonconvex quadratically constrained quadratic program (QCQP) with tractable convex reformulations. The vanilla TRS minimizes a quadratic function over the unit ball and has been well-studied in the literature [26,47,51]; for example, it is well known that the TRS has a second-order conic reformulation and can be solved efficiently [31]. Variants of the TRS have also been widely studied in the literature; see, e.g., the studies in [3,4,11,12,15–17,31,35,36,46,54,57,59,60,62–65]. Many of the aforementioned variants have been proven to be polynomial-time solvable to ϵ-optimality [11], but explicit semidefinite representable convex reformulations are known only for a few special cases such as the two-sided generalized TRS [49] and the TRS with non-intersecting linear constraints [17].

Recently, Eltved and Burer [21] consider the following variant, where a lower bound and an additional second-order cone constraint is added to the TRS:

$$\begin{aligned} \min \quad & \mathbf{x}^T H \mathbf{x} + 2\mathbf{g}^T \mathbf{x} \\ \text{s.t.} \quad & r \leq \|\mathbf{x}\| \leq R \\ & \|\mathbf{x} - \mathbf{c}\| \leq \mathbf{b}^T \mathbf{x} - a \end{aligned} \tag{2.1}$$

Here, $H \in \mathcal{S}^n$ is a real symmetric matrix, $\mathbf{x}, \mathbf{g}, \mathbf{c}, \mathbf{b} \in \mathbb{R}^n$, $a \in \mathbb{R}$, and $r, R \in \mathbb{R}_+$. The major novelty in [21] is that the authors construct nonnegative quartic expressions over the feasible region, which lead to a class of polynomial-time separable valid inequalities in the lifted space. Numerical results show that the valid inequalities are effective in reducing the gap of the strongest semidefinite programming (SDP) relaxations in the literature, especially in low dimensions. The authors also pay particular attention to two special cases of the problem. One case is when $r = 0$ and $\mathbf{b} = \mathbf{0}$, and the other is when $r = a = 0$ and $\mathbf{c} = \mathbf{0}$. The first case is a special case of the widely studied two-trust-region subproblem (TTRS). The second case is worth studying because that when $\mathbf{b} = \mathbf{e}$ (namely, $\mathbf{b}$ is the vector of all ones) and $R = 1$, its feasible region is a superset of the intersection of the unit ball and the nonnegative orthant, i.e., $\{\, \mathbf{x} \in \mathbb{R}^n \mid \|\mathbf{x}\| \leq 1,\, \mathbf{x} \geq \mathbf{0} \,\}$. Valid inequalities of the set can lead to cuts for the completely positive cone. Based on great numerical results, the authors conjecture that their valid inequalities will help to build an exact SDP reformulation of the second case when $n = 2$.

Motivated by the aforementioned encouraging numerical results and the conjecture made in [21], in this paper we focus on the two special cases mentioned above with the goal of developing their exact SDP reformulations. To our knowledge, no exact SDP reformulations for these two cases have been discovered in the literature; the nonnegative quartic expression technique and the associated polynomial-time separable valid inequalities in [21] are currently the strongest relaxation approach for solving both problems. Note that although the feasible regions in both cases are convex, the problems are still challenging due to the nonconvexity of the quadratic objective functions. However, with the great numerical results in [21], we are encouraged to think that certain exact SDP reformulation might be formed for these special cases.

Our research goal is achieved through a different (but possibly simpler) path from the nonnegative quartic expression approach in [21]. Specifically, the main ingredient of our study in this note is the partitions of the feasible regions and a convex hull result of disjunctions. Note that the idea of partitioning the feasible region of TRS variants has appeared previously in the literature. As far as we are aware of, the idea was first mentioned in [57] for feasible regions defined by two quadratic inequalities with the same Hessian. In [2], the authors solved quadratic programs with two ball constraints by partitioning the problem into two extended TRS. The idea was also used to develop a branch-and-bound algorithm for quadratic programs with ball and linear constraints in [1]. Most recently, Anstreicher proposed to solve the TTRS by branching the feasible region with eigenvector-based linear constraints [5]. However, all the above results focus solely from the perspective of developing polynomial-time solvable algorithms. The main difference between our paper and all the above results is that we focus not only on globally solving the problems but also on convexification of the problems. In particular, we provide a semidefinite representation of the lifted convex hull of the feasible region. The ability to solve the problems is an immediate consequence of the convexification.

Our contribution in this note is described in detail below. For future reference purposes, we restate the two special cases of (2.1) that we study in this paper (Without loss of generality, we assume that $R = 1$ in (2.1)):

$$\begin{aligned} \min \quad & \mathbf{x}^T H \mathbf{x} + 2\mathbf{g}^T \mathbf{x} \\ \text{s.t.} \quad & \|\mathbf{x}\| \leq 1 \\ & \|\mathbf{x} - \mathbf{c}\| \leq a \end{aligned} \tag{2.2}$$

and

$$\begin{aligned} \min \quad & \mathbf{x}^T H \mathbf{x} + 2\mathbf{g}^T \mathbf{x} \\ \text{s.t.} \quad & \|\mathbf{x}\| \leq 1 \\ & \|\mathbf{x}\| \leq \mathbf{b}^T \mathbf{x} - a. \end{aligned} \tag{2.3}$$

In the sequel, the former problem (2.2) will be referred to as the one with two ball constraints; the latter (2.3) will be referred to as the one with a ball and a second-order cone representable constraint. For each problem, we provide an exact convex reformulation through a semidefinite representation of the lifted convex hull of the feasible region. The representations are based on partitions of the feasible regions and a convex hull result of disjunctions.

The paper is organized as follows. In Section 2.2, we introduce the lifted convex hull and related properties. In Section 2.3, we derive the semidefinite representable reformulations of the two problems. A separation algorithm and computational results are presented in Section 2.4.

Notation. We use $\mathcal{S}^n$ and $\mathcal{S}^n_+$ to represent the sets of $n \times n$ real symmetric matrices and real symmetric positive semidefinite matrices, respectively. For $A, B \in \mathcal{S}^n$, the relation $A \succeq B$ holds if and only if $A - B \in \mathcal{S}^n_+$, and the Frobenious product of A and B is defined as $A \bullet B = \text{tr}(AB)$, where $\text{tr}(\cdot)$ is the trace of a matrix. The Kronecker product of A and B is denoted by $A \otimes B$. For a nonempty set $S \subseteq \mathbb{R}^n$, the convex hull of S is denoted by

$\mathrm{conv}(S)$.

2.2 The lifted convex hull

To begin, consider a general QCQP:

$$\begin{aligned} \inf \quad & \mathbf{x}^T H\mathbf{x} + 2\mathbf{g}^T\mathbf{x} \\ \text{s.t.} \quad & \mathbf{x} \in \mathcal{F}, \end{aligned} \tag{2.4}$$

where $H \in \mathcal{S}^n$ and $\mathcal{F} \subseteq \mathbb{R}^n$ is a nonempty closed set. The problem can be equivalently lifted to

$$\begin{aligned} \inf \quad & H \bullet X + 2\mathbf{g}^T\mathbf{x} \\ \text{s.t.} \quad & X = \mathbf{x}\mathbf{x}^T \\ & \mathbf{x} \in \mathcal{F} \end{aligned} \tag{2.5}$$

with variables $(\mathbf{x}, X)$ in the space of $\mathbb{R}^n \times \mathcal{S}^n$. Moreover, it is shown (e.g. in [20, 43]) that (1) is equivalent to

$$\begin{aligned} \inf \quad & H \bullet X + 2\mathbf{g}^T\mathbf{x} \\ \text{s.t.} \quad & (x, X) \in \mathcal{C}(\mathcal{F}), \end{aligned} \tag{2.6}$$

where

$$\mathcal{C}(\mathcal{F}) := \mathrm{conv}\left\{\, (\mathbf{x}, \mathbf{x}\mathbf{x}^T) \mid \mathbf{x} \in \mathcal{F} \,\right\}. \tag{2.7}$$

We refer to $\mathcal{C}(\mathcal{F})$ as the lifted convex hull in this paper. The analysis on $\mathcal{C}(\mathcal{F})$ has been critical in several previous QCQP studies in the literature; see, e.g., [15, 38, 63] and the references within. The derivation throughout our paper also relies heavily on the characterization of $\mathcal{C}(\mathcal{F})$. Note that when $\mathcal{F}$ is compact, so is the lifted convex hull $\mathcal{C}(\mathcal{F})$, and optimal solutions of (2.4)-(2.6) can be attained.

The main results of this paper are explicit semidefinite representations of $\mathcal{C}(\mathcal{F})$ for (2.2) and (2.3). In this section, we introduce preliminary results related to $\mathcal{C}(\mathcal{F})$. The following lemma characterizes the lifted convex hull of the union of two sets.

Lemma 3. *Let $\mathcal{F}_1$ and $\mathcal{F}_2$ be two nonempty closed sets in $\mathbb{R}^n$. We have*

$$\begin{aligned}\mathcal{C}(\mathcal{F}_1 \cup \mathcal{F}_2) &= \mathrm{conv}(\mathcal{C}(\mathcal{F}_1) \cup \mathcal{C}(\mathcal{F}_2)) \\ &= \left\{ (\mathbf{x}, X) \;\middle|\; \begin{array}{l} \exists\ \lambda \in [0,1],\ (\mathbf{x}_1, X_1) \in \mathcal{C}(\mathcal{F}_1),\ (\mathbf{x}_2, X_2) \in \mathcal{C}(\mathcal{F}_2) \\ \textit{such that } (\mathbf{x}, X) = \lambda(\mathbf{x}_1, X_1) + (1-\lambda)(\mathbf{x}_2, X_2) \end{array} \right\}.\end{aligned}$$

Proof. For any nonempty sets A and B in $\mathbb{R}^n$, it is clear that $\mathrm{conv}(A \cup B) = \mathrm{conv}(\mathrm{conv}(A) \cup \mathrm{conv}(B))$. Let $A = \{ (\mathbf{x}, \mathbf{x}\mathbf{x}^T) \mid \mathbf{x} \in \mathcal{F}_1 \}$ and $B = \{ (\mathbf{x}, \mathbf{x}\mathbf{x}^T) \mid \mathbf{x} \in \mathcal{F}_2 \}$. Then,

$$\begin{aligned}\mathcal{C}(\mathcal{F}_1 \cup \mathcal{F}_2) &= \mathrm{conv}\,\{ (\mathbf{x}, \mathbf{x}\mathbf{x}^T) \mid \mathbf{x} \in \mathcal{F}_1 \cup \mathcal{F}_2 \} = \mathrm{conv}(A \cup B) \\ &= \mathrm{conv}(\mathrm{conv}(A) \cup \mathrm{conv}(B)) = \mathrm{conv}(\mathcal{C}(\mathcal{F}_1) \cup \mathcal{C}(\mathcal{F}_2)).\end{aligned}$$

The second equation in the statement follows directly from the definition of the convex hull. □

Remark. *Lemma 3 can also be derived from the perspective of the cone of nonnegative quadratic functions. See [37]. The above proof takes a basic approach without considering propositions of the dual cones.*

The structure of $\mathcal{C}(\mathcal{F})$ has been studied in the literature for specially structured $\mathcal{F}$ (citations). In the following, we list two results related to our derivation in Section 2.3. The first proposition is related to the intersection of a halfspace and a ball.

Proposition 4 ([57], Theorem 3)**.** *For* $\mathcal{F} = \{\, \mathbf{x} \in \mathbb{R}^n \mid \|\mathbf{x} - \mathbf{c}\| \leq a, \mathbf{p}^T\mathbf{x} + q \geq 0 \,\}$*, where* $\mathbf{p} \in \mathbb{R}^n$ *and* $q \in \mathbb{R}$*,*

$$\mathcal{C}(\mathcal{F}) = \left\{ (\mathbf{x}, X) \;\middle|\; \begin{array}{l} \mathrm{tr}(X) - 2\mathbf{c}^T\mathbf{x} + \mathbf{c}^T\mathbf{c} - a^2 \leq 0, \\ \|X\mathbf{p} + q\mathbf{x} - \mathbf{c}\mathbf{p}^T\mathbf{x} - q\mathbf{c}\| \leq a(\mathbf{p}^T\mathbf{x} + q),\; X \succeq \mathbf{x}\mathbf{x}^T \end{array} \right\}.$$

We briefly explain the construction of the constraints in $\mathcal{C}(\mathcal{F})$. The first constraint is built by squaring both sides of $\|\mathbf{x} - \mathbf{c}\| \leq a$ and lifting $\mathbf{x}\mathbf{x}^T$ to X. The second constraint is obtained by multiplying both sides of $\|\mathbf{x} - \mathbf{c}\| \leq a$ by the nonnegative quantity $(\mathbf{p}^T\mathbf{x} + q)$ and lifting $\mathbf{x}\mathbf{x}^T$ to X. The last constraint is a semidefinite relaxation of the nonconvex constraint $X = \mathbf{x}\mathbf{x}^T$.

The next proposition considers the intersection of a halfspace and a second-order cone representable set that share certain special structure (in terms of the linear term in their descriptions).

Proposition 5 ([37], Corollary 2)**.** *For* $\mathcal{F} = \{\, \mathbf{x} \in \mathbb{R}^n \mid \|\mathbf{x}\| \leq \mathbf{p}^T\mathbf{x} + q \leq s \,\}$*, where* $\mathbf{p} \in \mathbb{R}^n$ *and* $q, s \in \mathbb{R}$*,*

$$\mathcal{C}(\mathcal{F}) = \left\{ (\mathbf{x}, X) \;\middle|\; \begin{array}{l} (I - \mathbf{p}\mathbf{p}^T) \bullet X - 2q\mathbf{p}^T\mathbf{x} - q^2 \leq 0,\;\; \mathbf{p}^T\mathbf{x} + q - s \leq 0, \\ \|X\mathbf{p} + (q - s)\mathbf{x}\| \leq -\mathbf{p}\mathbf{p}^T \bullet X + (s - 2q)\mathbf{p}^T\mathbf{x} - q(q - s),\; X \succeq \mathbf{x}\mathbf{x}^T \end{array} \right\}.$$

The construction of the constraints in $\mathcal{C}(\mathcal{F})$ in Proposition 5 is similar to that in Proposition 4. The first constraint is built by squaring both sides of $\|\mathbf{x}\| \leq \mathbf{p}^T\mathbf{x} + q$ and lifting $\mathbf{x}\mathbf{x}^T$ to X. The second constraint is directly inherited from $\mathcal{F}$. The third constraint is obtained by multiplying both sides of $\|\mathbf{x}\| \leq \mathbf{p}^T\mathbf{x} + q$ by the nonnegative quantity $(s - q - \mathbf{p}^T\mathbf{x})$ and lifting $\mathbf{x}\mathbf{x}^T$ to X.

2.3 Semidefinite reformulations

In this section, we derive semidefinite reformulations of (2.2) and (2.3). The key idea is to partition the feasible region $\mathcal{F}$ of each problem as $\mathcal{F} = \mathcal{F}_1 \cup \mathcal{F}_2$, where $\mathcal{C}(\mathcal{F}_1)$ and

$\mathcal{C}(\mathcal{F}_2)$ are known.

2.3.1 Two ball constraints

Let $\mathcal{F}_{TB} = \{\, \mathbf{x} \in \mathbb{R}^n \mid \ \|\mathbf{x}\| \le 1,\ \|\mathbf{x}-\mathbf{c}\| \le a \,\}$ be the feasible region of (2.2). We ignore the trivial cases when $\mathcal{F}_{TB}$ is empty, or not full-dimensional (when $\mathcal{F}_{TB}$ is a singleton), or one ball is contained in the other (when $\mathcal{F}_{TB}$ is a ball and problem (2.2) reduces to the TRS). In the nontrivial cases, observe that the intersection of $\{\, \mathbf{x} \in \mathbb{R}^n \mid \|\mathbf{x}\| \le 1 \,\}$ and $\{\, \mathbf{x} \in \mathbb{R}^n \mid \|\mathbf{x}-\mathbf{c}\| \le a \,\}$ is an $(n-1)$-dimensional sphere. Therefore, the following partition is possible for $\mathcal{F}_{TB}$.

Lemma 6 ([2], Theorem 2.3)**.** *For the nontrivial cases of problem* (2.2) *with two ball constraints, $\mathcal{F}_{TB} = \mathcal{F}_1 \cup \mathcal{F}_2$, where*

$$\mathcal{F}_1 = \{\, \mathbf{x} \in \mathbb{R}^n \mid \|\mathbf{x}\| \le 1,\ -2\mathbf{c}^T\mathbf{x} + (1 + \mathbf{c}^T\mathbf{c} - a^2) \le 0 \,\},$$

$$\mathcal{F}_2 = \{\, \mathbf{x} \in \mathbb{R}^n \mid \|\mathbf{x}-\mathbf{c}\| \le a,\ -2\mathbf{c}^T\mathbf{x} + (1 + \mathbf{c}^T\mathbf{c} - a^2) \ge 0 \,\}.$$

With Lemma 3 and Proposition 4, $\mathcal{C}(\mathcal{F}_{TB})$ has the following semidefinite representation.

Proposition 7. *For the nontrivial cases of problem* (2.2) *with two ball constraints, we have*

$$\mathcal{C}(\mathcal{F}_{TB}) = \left\{ (\mathbf{x}, X) \;\middle|\; \begin{array}{l} \exists\, \lambda \in [0,1],\ (\mathbf{y}_1, Y_1),\ (\mathbf{y}_2, Y_2) \in \mathbb{R}^n \times \mathcal{S}^n \ \textit{such that} \\ (\mathbf{x}, X) = (\mathbf{y}_1, Y_1) + (\mathbf{y}_2, Y_2), \\ \operatorname{tr}(Y_1) \le \lambda,\ \|2Y_1\mathbf{c} - q\mathbf{y}_1\| \le 2\mathbf{c}^T\mathbf{y}_1 - q\lambda, \\ \operatorname{tr}(Y_2) - 2\mathbf{c}^T\mathbf{y}_2 + (\mathbf{c}^T\mathbf{c} - a^2)(1-\lambda) \le 0, \\ \|2Y_2\mathbf{c} - q\mathbf{y}_2 - 2\mathbf{c}\mathbf{c}^T\mathbf{y}_2 + q\mathbf{c}(1-\lambda)\| \le a(-2\mathbf{c}^T\mathbf{y}_2 + q(1-\lambda)), \\ \begin{pmatrix} \lambda & \mathbf{y}_1^T \\ \mathbf{y}_1 & Y_1 \end{pmatrix} \succeq 0,\ \begin{pmatrix} 1-\lambda & \mathbf{y}_2^T \\ \mathbf{y}_2 & Y_2 \end{pmatrix} \succeq 0 \end{array} \right\},$$

where $q = 1 + \mathbf{c}^T\mathbf{c} - a^2$.

Proof. Denote the set on the right side of the equation by S. Recalling the sets $\mathcal{F}_1$ and $\mathcal{F}_2$ defined in Lemma 6 applying Proposition 4 we have

$$\mathcal{C}(\mathcal{F}_1) = \{ (\mathbf{x}_1, X_1) \mid \operatorname{tr}(X_1) \le 1,\ \|2X_1\mathbf{c} - q\mathbf{x}_1\| \le 2\mathbf{c}^T\mathbf{x}_1 - q,\ X_1 \succeq \mathbf{x}_1\mathbf{x}_1^T \},$$
$$\mathcal{C}(\mathcal{F}_2) = \left\{ (\mathbf{x}_2, X_2) \;\middle|\; \begin{array}{l} \|2X_2\mathbf{c} - q\mathbf{x}_2 - 2\mathbf{c}\mathbf{c}^T\mathbf{x}_2 - q\mathbf{c}\| \le a(-2\mathbf{c}^T\mathbf{x}_2 + q), \\ \operatorname{tr}(X_2) - 2\mathbf{c}^T\mathbf{x}_2 + \mathbf{c}^T\mathbf{c} - a^2 \le 0,\ X_2 \succeq \mathbf{x}_2\mathbf{x}_2^T \end{array} \right\}.$$

Note that for any $\lambda \in (0, 1]$, $(\mathbf{x}_1, X_1) \in \mathcal{C}(\mathcal{F}_1)$ if and only if $(\mathbf{y}_1, Y_1) = \lambda(\mathbf{x}_1, X_1)$ satisfies

$$\operatorname{tr}(Y_1) \le \lambda,\ \|2Y_1\mathbf{c} - q\mathbf{y}_1\| \le 2\mathbf{c}^T\mathbf{y}_1 - q\lambda,\ \text{ and } \lambda Y_1 \succeq \mathbf{y}_1\mathbf{y}_1^T.$$

Similarly, for any $\lambda \in [0, 1)$, $(\mathbf{x}_2, X_2) \in \mathcal{C}(\mathcal{F}_2)$ if and only if $(\mathbf{y}_2, Y_2) = (1-\lambda)(\mathbf{x}_2, X_2)$ satisfies

$$\begin{aligned} &\|2Y_2\mathbf{c} - q\mathbf{y}_2 - 2\mathbf{c}\mathbf{c}^T\mathbf{y}_2 - (1-\lambda)q\mathbf{c}\| \le a(-2\mathbf{c}^T\mathbf{y}_2 + q(1-\lambda)), \\ &\operatorname{tr}(Y_2) - 2\mathbf{c}^T\mathbf{y}_2 + (1-\lambda)(\mathbf{c}^T\mathbf{c} - a^2) \le 0,\ \text{ and } (1-\lambda)Y_2 \succeq \mathbf{y}_2\mathbf{y}_2^T. \end{aligned}$$

For any $(\mathbf{x}, X) \in \mathcal{C}(\mathcal{F}_{TB})$, by Lemma 3, there exist $\lambda \in [0, 1]$, $(\mathbf{x}_1, X_1) \in \mathcal{C}(\mathcal{F}_1)$ and $(\mathbf{x}_2, X_2) \in \mathcal{C}(\mathcal{F}_2)$ such that $(\mathbf{x}, X) = \lambda(\mathbf{x}_1, X_1) + (1-\lambda)(\mathbf{x}_2, X_2)$. Let

$$(\mathbf{y}_1, Y_1) = \begin{cases} \lambda(\mathbf{x}_1, X_1) & \text{if } \lambda > 0 \\ (\mathbf{0}, 0) & \text{if } \lambda = 0 \end{cases} \quad \text{and} \quad (\mathbf{y}_2, Y_2) = \begin{cases} (1-\lambda)(\mathbf{x}_2, X_2) & \text{if } \lambda < 1 \\ (\mathbf{0}, 0) & \text{if } \lambda = 1. \end{cases}$$

Then $(\mathbf{x}, X) \in S$ as $(\mathbf{x}, X, \lambda, \mathbf{y}_1, Y_1, \mathbf{y}_2, Y_2)$ satisfies the constraints in S due to the above derivation. On the other hand, for any $(\mathbf{x}, X) \in S$, there exists $(\lambda, \mathbf{y}_1, Y_1, \mathbf{y}_2, Y_2)$ satisfying the constraints in S. Let

$$(\mathbf{x}_1, X_1) = \begin{cases} \frac{1}{\lambda}(\mathbf{y}_1, Y_1) & \text{if } \lambda > 0 \\ \text{any point in } \mathcal{C}(\mathcal{F}_1) & \text{if } \lambda = 0 \end{cases} \text{ and } (\mathbf{x}_2, X_2) = \begin{cases} \frac{1}{1-\lambda}(\mathbf{y}_2, Y_2) & \text{if } \lambda < 1 \\ \text{any point in } \mathcal{C}(\mathcal{F}_2) & \text{if } \lambda = 1. \end{cases}$$

Then it is clear that $(\mathbf{x}, X) \in \mathcal{C}(\mathcal{F}_{TB})$. $\square$

2.3.2 A ball and a second-order cone representable constraint

Let $\mathcal{F}_{SOC} = \{ \mathbf{x} \in \mathbb{R}^n \mid \|\mathbf{x}\| \leq 1, \ \|\mathbf{x}\| \leq \mathbf{b}^T\mathbf{x} - a \}$ be the feasible region of (2.3). We make the following assumptions to avoid trivial cases.

Assumption 1. *There exist* $\tilde{\mathbf{x}}, \hat{\mathbf{x}}, \bar{\mathbf{x}} \in \mathbb{R}^n$ *such that*

$$\|\tilde{\mathbf{x}}\| \leq 1, \ \|\tilde{\mathbf{x}}\| \leq \mathbf{b}^T\tilde{\mathbf{x}} - a, \tag{2.8}$$

$$\|\bar{\mathbf{x}}\| \leq 1, \ \|\bar{\mathbf{x}}\| > \mathbf{b}^T\bar{\mathbf{x}} - a, \tag{2.9}$$

$$\|\hat{\mathbf{x}}\| > 1, \ \|\hat{\mathbf{x}}\| \leq \mathbf{b}^T\hat{\mathbf{x}} - a. \tag{2.10}$$

Assumption 1 guarantees that $\{ \mathbf{x} \in \mathbb{R}^n \mid \|\mathbf{x}\| \leq 1 \}$ and $\{ \mathbf{x} \in \mathbb{R}^n \mid \|\mathbf{x}\| \leq \mathbf{b}^T\mathbf{x} - a \}$ intersect and neither of the sets contains the other. When Assumption 1 is violated, $\mathcal{F}_{SOC}$ is either empty, a ball (when problem (2.3) reduces to the TRS), or a set defined by a second-order cone constraint (when problem (2.3) can be handled by Corollary 1 in [37]). It is clear that $\mathcal{F}_{SOC} = \mathcal{F}_1 \cup \mathcal{F}_2$, where

$$\begin{aligned} \mathcal{F}_1 &= \{ \mathbf{x} \in \mathbb{R}^n \mid \|\mathbf{x}\| \leq 1 \leq \mathbf{b}^T\mathbf{x} - a \}, \\ \mathcal{F}_2 &= \{ \mathbf{x} \in \mathbb{R}^n \mid \|\mathbf{x}\| \leq \mathbf{b}^T\mathbf{x} - a \leq 1 \}. \end{aligned} \tag{2.11}$$

Under Assumption 1, we can show that both $\mathcal{F}_1$ and $\mathcal{F}_2$ are nonempty.

Lemma 8. *Under Assumption 1, there exists* $\mathbf{x}_0 \in \mathbb{R}^n$ *such that* $\|\mathbf{x}_0\| \leq 1 = \mathbf{b}^T\mathbf{x}_0 - a$.

Proof. If $\mathbf{b}^T\tilde{\mathbf{x}} - a = 1$, then $\mathbf{x}_0 = \tilde{\mathbf{x}}$ is such a point. If $\mathbf{b}^T\tilde{\mathbf{x}} - a > 1$, since $\|\bar{\mathbf{x}}\| \leq 1$ and $\mathbf{b}^T\bar{\mathbf{x}} - a < 1$, we can choose $\mathbf{x}_0$ as the convex combination of $\tilde{\mathbf{x}}$ and $\bar{\mathbf{x}}$ such that $\mathbf{b}^T\mathbf{x}_0 - a = 1$. If $\mathbf{b}^T\tilde{\mathbf{x}} - a < 1$, since $\|\hat{\mathbf{x}}\| \leq \mathbf{b}^T\hat{\mathbf{x}} - a$ and $\mathbf{b}^T\hat{\mathbf{x}} - a > 1$, we can choose $\mathbf{x}_0$ as the convex combination of $\tilde{\mathbf{x}}$ and $\hat{\mathbf{x}}$ such that $\mathbf{b}^T\mathbf{x}_0 - a = 1$. $\square$

With Lemma 3 and Propositions 4 and 5, $\mathcal{C}(\mathcal{F}_{SOC})$ has the following semidefinite representation.

Proposition 9. *Under Assumption 1,*

$$\mathcal{C}(\mathcal{F}_{SOC}) = \left\{ (\mathbf{x}, X) \;\middle|\; \begin{array}{l} \exists\ \lambda \in [0,1],\ (\mathbf{y}_1, Y_1),\ (\mathbf{y}_2, Y_2) \in \mathbb{R}^n \times \mathcal{S}^n \textit{ such that} \\ (\mathbf{x}, X) = (\mathbf{y}_1, Y_1) + (\mathbf{y}_2, Y_2), \\ \mathrm{tr}(Y_1) \le \lambda,\ \ \|Y_1\mathbf{b} - (a+1)\mathbf{y}_1\| \le \mathbf{b}^T\mathbf{y}_1 - (a+1)\lambda, \\ (I - \mathbf{b}\mathbf{b}^T) \bullet Y_2 + 2a\mathbf{b}^T\mathbf{y}_2 - a^2(1-\lambda) \le 0, \\ \mathbf{b}^T\mathbf{y}_2 - (a+1)(1-\lambda) \le 0, \\ \|Y_2\mathbf{b} - (a+1)\mathbf{y}_2\| \le -\mathbf{b}\mathbf{b}^T \bullet Y_2 + (1+2a)\mathbf{b}^T\mathbf{y}_2 - a(1+a)(1-\lambda), \\ \begin{pmatrix} \lambda & \mathbf{y}_1^T \\ \mathbf{y}_1 & Y_1 \end{pmatrix} \succeq 0,\ \begin{pmatrix} 1-\lambda & \mathbf{y}_2^T \\ \mathbf{y}_2 & Y_2 \end{pmatrix} \succeq 0 \end{array} \right\}.$$

Proof. The proof is similar to that of Proposition 7. Denote the set on the right side of the equation by S. Also, let $\mathcal{F}_1$ and $\mathcal{F}_2$ be the sets defined in (2.11). By Propositions 4 and 5,

$$\mathcal{C}(\mathcal{F}_1) = \{\, (\mathbf{x}_1, X_1) \mid \mathrm{tr}(X_1) \le 1,\ \|X_1\mathbf{b} - (a+1)\mathbf{x}_1\| \le \mathbf{b}^T\mathbf{x}_1 - a - 1,\ X_1 \succeq \mathbf{x}_1\mathbf{x}_1^T \,\},$$
$$\mathcal{C}(\mathcal{F}_2) = \left\{ (\mathbf{x}_2, X_2) \;\middle|\; \begin{array}{l} \|X_2\mathbf{b} - (a+1)\mathbf{x}_2\| \le -\mathbf{b}\mathbf{b}^T \bullet X_2 + (2a+1)\mathbf{b}^T\mathbf{x}_2 - a(a+1), \\ (I - \mathbf{b}\mathbf{b}^T) \bullet X_2 + 2a\mathbf{b}^T\mathbf{x}_2 - a^2 \le 0,\ \mathbf{b}^T\mathbf{x}_2 - a - 1 \le 0,\ X_2 \succeq \mathbf{x}_2\mathbf{x}_2^T \end{array} \right\}.$$

For any $\lambda \in (0,1)$, $(\mathbf{x}_1, X_1) \in \mathcal{C}(\mathcal{F}_1)$ if and only if $(\mathbf{y}_1, Y_1) = \lambda(\mathbf{x}_1, X_1)$ satisfies

$$\mathrm{tr}(Y_1) \le \lambda,\ \|Y_1\mathbf{b} - (a+1)\mathbf{y}_1\| \le \mathbf{b}^T\mathbf{y}_1 - (a+1)\lambda,\ \text{ and } \lambda Y_1 \succeq \mathbf{y}_1\mathbf{y}_1^T.$$

Similarly, $(\mathbf{x}_2, X_2) \in \mathcal{C}(\mathcal{F}_2)$ if and only if $(\mathbf{y}_2, Y_2) = (1-\lambda)(\mathbf{x}_2, X_2)$ satisfies

$$(I - \mathbf{b}\mathbf{b}^T) \bullet Y_2 + 2a\mathbf{b}^T\mathbf{y}_2 - a^2(1-\lambda) \le 0,\ \mathbf{b}^T\mathbf{y}_2 - (a+1)(1-\lambda) \le 0,$$
$$\|Y_2\mathbf{b} - (a+1)\mathbf{y}_2\| \le -\mathbf{b}\mathbf{b}^T \bullet Y_2 + (2a+1)\mathbf{b}^T\mathbf{y}_2 - a(a+1)(1-\lambda)),\ \text{ and } (1-\lambda)Y_2 \succeq \mathbf{y}_2\mathbf{y}_2^T.$$

The rest of the proof is the same as the proof of Proposition 7 with $\mathcal{F}_{TB}$ replaced by $\mathcal{F}_{SOC}$. □

2.3.3 Discussion on the trivial cases

Although Propositions 7 and 9 are derived for the nontrivial cases, they hold in the trivial cases as well. In this section, we explain how the set on the right side of the equation in Proposition 9, denoted by S, is equal to $\mathcal{C}(\mathcal{F}_{SOC})$ when Assumption 1 fails. The trivial cases for the two-ball problem can be analyzed in the same manner.

First, if there is no $\tilde{x} \in \mathbb{R}^n$ satisfying (2.8), then $\mathcal{F} = \mathcal{F}_1 = \mathcal{F}_2 = \emptyset$, where $\mathcal{F}_1$ and $\mathcal{F}_2$ are defined in (2.11). We show that $S = \emptyset = \mathcal{C}(\mathcal{F}_{SOC})$ by contradiction. If $S \neq \emptyset$, there exist $\lambda \in [0, 1], (\mathbf{y}_1, Y_1), (\mathbf{y}_2, Y_2)$ and $(\mathbf{x}, X)$ such that the constraints in S are satisfied. If $\lambda > 0$, it is easy to check that $(\mathbf{y}_1, Y_1)/\lambda \in \mathcal{C}(\mathcal{F}_1)$, which contradicts to $\mathcal{F}_1 = \emptyset$. If $\lambda = 0$, then $(\mathbf{y}_2, Y_2) \in \mathcal{C}(\mathcal{F}_2)$, which contradicts to $\mathcal{F}_2 = \emptyset$.

Next, we consider the case when (2.8) is satisfied, but (2.9) or (2.10) is not. We start from the following lemma.

Lemma 10. *Suppose that* (2.8) *is satisfied. If there is no* $\bar{x} \in \mathbb{R}^n$ *satisfying* (2.9)*, then* $\mathcal{F}_2 \subseteq \mathcal{F}_1$*. If there is no* $\hat{x} \in \mathbb{R}^n$ *satisfying* (2.10)*, then* $\mathcal{F}_1 \subseteq \mathcal{F}_2$*.*

Proof. To prove the first statement, it suffices to show that $\mathbf{b}^T\mathbf{x} - a \geq 1$ for all $\mathbf{x} \in \mathcal{F}_2$. In fact, since (2.9) is violated, $\|\mathbf{x}\| \leq \mathbf{b}^T\mathbf{x} - a$ for all $\mathbf{x}$ such that $\|\mathbf{x}\| \leq 1$. Substituting $\mathbf{x}$ for $-\mathbf{b}/\|\mathbf{b}\|$, it is clear that $1 \leq -\|\mathbf{b}\| - a$. Therefore, for any $\mathbf{x} \in \mathcal{F}_2$, since $\|\mathbf{x}\| \leq 1$, we have $\mathbf{b}^T\mathbf{x} - a \geq -\|\mathbf{b}\| - a \geq 1$.

For the second statement, we first note that if (2.8) is satisfied and (2.10) is violated, one must have $\|\mathbf{b}\| < 1$ and $a \leq 0$. In fact, if $\|\mathbf{b}\| > 1$ or $a < 0$ and $\|\mathbf{b}\| = 1$, one can find an eigenvector of $I - \mathbf{b}\mathbf{b}^T$ such that $\{\, \mathbf{x} \in \mathbb{R}^n \mid \|\mathbf{x}\| \leq \mathbf{b}^T\mathbf{x} - a \,\}$ is unbounded along the direction of the eigenvector. In that case, $\{\, \mathbf{x} \in \mathbb{R}^n \mid \|\mathbf{x}\| \leq \mathbf{b}^T\mathbf{x} - a \,\}$ cannot be contained in the unit ball, and therefore, (2.10) is not violated. If $a > 0$ and $\|\mathbf{b}\| \leq 1$, it is easy to check that $\mathbf{b}^T\mathbf{x} - a < \mathbf{b}^T\mathbf{x} \leq \|\mathbf{x}\|$. That is, $\{\, \mathbf{x} \in \mathbb{R}^n \mid \|\mathbf{x}\| \leq \mathbf{b}^T\mathbf{x} - a \,\} = \emptyset$ and (2.8) is not satisfied. Therefore, we only need to consider the case when $\|\mathbf{b}\| < 1$ and $a \leq 0$.

To prove the second statement, it suffices to show that $\mathbf{b}^T\mathbf{x} - a \leq 1$ for all $\mathbf{x} \in \mathcal{F}_1$. Since (2.10) is violated, $\|\mathbf{x}\| \leq 1$ for all $\mathbf{x}$ such that $\|\mathbf{x}\| \leq \mathbf{b}^T\mathbf{x} - a$. Substituting $\mathbf{x}$

for $(a/(\|\mathbf{b}\|^2 - \|\mathbf{b}\|))\,\mathbf{b}$, we see that $\mathbf{b}^T(a/(\|\mathbf{b}\|^2 - \|\mathbf{b}\|))\,\mathbf{b} - a = a\|\mathbf{b}\|/(\|\mathbf{b}\| - 1) - a = a/(\|\mathbf{b}\| - 1) = \|(a/(\|\mathbf{b}\|^2 - \|\mathbf{b}\|))\,\mathbf{b}\|$. Therefore, $a/(\|\mathbf{b}\| - 1) \leq 1$. That is, $\|\mathbf{b}\| - 1 \leq a$. Consequently, for any $\mathbf{x} \in \mathcal{F}_1$, $\|\mathbf{x}\| \leq 1$ and $\mathbf{b}^T\mathbf{x} - a \leq \|\mathbf{b}\| - a \leq 1$. $\square$

In the case when (2.8) is satisfied but (2.9) is not, Lemma 10 indicates that $\mathcal{C}(\mathcal{F}_2) \subseteq \mathcal{C}(\mathcal{F}_1)$. Therefore, $\mathcal{C}(\mathcal{F}_{SOC}) = \mathcal{C}(\mathcal{F}_1)$ by Lemma 3. We prove $S = \mathcal{C}(\mathcal{F}_{SOC})$ by showing $S = \mathcal{C}(\mathcal{F}_1)$. For any $(\mathbf{x}, X) \in S$, there exist $\lambda \in [0,1], (\mathbf{y}_1, Y_1)$ and $(\mathbf{y}_2, Y_2)$ such that the constraints in S are satisfied. If $\lambda = 1$, the constraints imply $(\mathbf{y}_2, Y_2) = (0,0)$, which indicates $(\mathbf{x}, X) = (\mathbf{y}_1, Y_1) \in \mathcal{C}(\mathcal{F}_1)$. If $\lambda < 1$, then $(\mathbf{y}_2, Y_2)/(1-\lambda) \in \mathcal{C}(\mathcal{F}_2) \subseteq \mathcal{C}(\mathcal{F}_1)$. Since $(\mathbf{y}_1, Y_1)/\lambda \in \mathcal{C}(\mathcal{F}_1)$, as a convex combination of $(\mathbf{y}_1, Y_1)/\lambda$ and $(\mathbf{y}_2, Y_2)/(1-\lambda)$, $(\mathbf{x}, X)$ is also in $\mathcal{C}(\mathcal{F}_1)$. Therefore, $S \subseteq \mathcal{C}(\mathcal{F}_1)$. On the other hand, for any $(\mathbf{x}, X) \in \mathcal{C}(\mathcal{F}_1)$, let $(\mathbf{y}_1, Y_1) = (\mathbf{x}, X)$, $\lambda = 1$, and $(\mathbf{y}_2, Y_2) = (0,0)$. The constraints in S are satisfied by $(\mathbf{x}, X), \lambda, (\mathbf{y}_1, Y_1)$ and $(\mathbf{y}_2, Y_2)$. Therefore, $\mathcal{C}(\mathcal{F}_1) \subseteq S$, and thus $S = \mathcal{C}(\mathcal{F}_1)$.

With a similar argument, one can show that when (2.8) is satisfied but (2.10) is not, $S = \mathcal{C}(\mathcal{F}_2) = \mathcal{C}(\mathcal{F}_{SOC})$. To conclude, the semidefinite representation in Proposition 9 is still valid when Assumption 1 fails.

2.4 Separation and computational results

The convex hull results we derive in Section 2.3 are presented as projections of convex sets in a higher-dimensional space. Due to the nonlinear feature of the sets, it is challenging to find explicit expressions of the projections in the space of $(\mathbf{x}, X)$. As mentioned in the introduction, valid inequalities of $\mathcal{C}(\mathcal{F})$ in the $(\mathbf{x}, X)$-space are of interest in certain applications such as finding valid inequalities for the completely positive cone. In this section, we show how to generate valid inequalities for the lifted convex hull in the $(\mathbf{x}, X)$-space as needed. In particular, for any given $(\hat{\mathbf{x}}, \hat{X})$ not in the lifted convex hull, a separating hyperplane can be found by solving an SDP. We use problem (2.3) as an example to explain the idea, and the separation problem for (2.2) can be built in the same manner.

For a given $(\hat{\mathbf{x}}, \hat{X})$, we consider the following separation problem for $\mathcal{C}(\mathcal{F}_{SOC})$.

$$
\begin{aligned}
v^*(\hat{\mathbf{x}}, \hat{X}) := \inf \quad & u_1 + u_2 + u_3 + u_4 + u_5 + w_1 + w_2 \\
\text{s.t.} \quad & \begin{pmatrix} \lambda & \mathbf{y}_1^T \\ \mathbf{y}_1 & Y_1 \end{pmatrix} + \begin{pmatrix} \mu & \mathbf{y}_2^T \\ \mathbf{y}_2 & Y_2 \end{pmatrix} = \begin{pmatrix} 1 & \hat{\mathbf{x}}^T \\ \hat{\mathbf{x}} & \hat{X} \end{pmatrix} \\
& \mathrm{tr}(Y_1) \le \lambda + u_1 \\
& \|Y_1 \mathbf{b} - (a+1)\mathbf{y}_1\| \le \mathbf{b}^T \mathbf{y}_1 - (a+1)\lambda + u_2 \\
& (I - \mathbf{b}\mathbf{b}^T) \bullet Y_2 + 2a\mathbf{b}^T \mathbf{y}_2 - a^2 \mu \le u_3 \\
& \|Y_2 \mathbf{b} - (a+1)\mathbf{y}_2\| \le -\mathbf{b}\mathbf{b}^T \bullet Y_2 + (1+2a)\mathbf{b}^T \mathbf{y}_2 - a(1+a)\mu + u_4 \\
& \mathbf{b}^T \mathbf{y}_2 - (a+1)\mu \le u_5 \\
& \begin{pmatrix} \lambda & \mathbf{y}_1^T \\ \mathbf{y}_1 & Y_1 \end{pmatrix} + w_1 I_{n+1} \succeq 0 \\
& \begin{pmatrix} \mu & \mathbf{y}_2^T \\ \mathbf{y}_2 & Y_2 \end{pmatrix} + w_2 I_{n+1} \succeq 0 \\
& \lambda, \mu, u_1, u_2, u_3, u_4, u_5, w_1, w_2 \ge 0
\end{aligned}
\tag{2.12}
$$

Here, $\mathbf{u} \in \mathbb{R}^5$ and $\mathbf{w} \in \mathbb{R}^2$ are nonnegative artificial variables. The constraints in (2.12) are basically the constraints in Proposition 9 with artificial variables $\mathbf{u}$ and $\mathbf{w}$. (A new variable $\mu = 1 - \lambda$ is introduced for the ease of derivation of the dual problem.) Obviously, if $v^*(\hat{\mathbf{x}}, \hat{X}) = 0$ and $(\lambda, \mu, \mathbf{y}_1, Y_1, \mathbf{y}_2, Y_2, \mathbf{u}, \mathbf{w})$ is an optimal solution, then $\mathbf{u} = \mathbf{0}$, $\mathbf{w} = \mathbf{0}$, and $(\lambda, \mathbf{y}_1, Y_1, \mathbf{y}_2, Y_2)$ satisfies the constraints in $\mathcal{C}(\mathcal{F}_{SOC})$ together with $(\hat{\mathbf{x}}, \hat{X})$. Therefore, $(\hat{\mathbf{x}}, \hat{X}) \in \mathcal{C}(\mathcal{F}_{SOC})$. If $v^*(\hat{\mathbf{x}}, \hat{X}) > 0$, then such a $(\lambda, \mathbf{y}_1, Y_1, \mathbf{y}_2, Y_2)$ does not exist and $(\hat{\mathbf{x}}, \hat{X}) \notin \mathcal{C}(\mathcal{F}_{SOC})$.

Consider the dual problem of (2.12). Let $Z \in \mathcal{S}^{n+1}$ be the dual variable associated with the equality constraint. The dual problem of (2.12) can be represented as

$$d^*(\hat{\mathbf{x}}, \hat{X}) := \sup \quad \begin{pmatrix} 1 & \hat{\mathbf{x}}^T \\ \hat{\mathbf{x}} & \hat{X} \end{pmatrix} \bullet Z \tag{2.13}$$
$$\text{s.t.} \quad Z \in \mathcal{D},$$

where $\mathcal{D}$ is a semidefinite representable set independent of the choice of $(\hat{\mathbf{x}}, \hat{X})$. (We omit the specific expression of $\mathcal{D}$ here as it is easy but tedious to show.) Since (2.12) clearly satisfies the Slater's condition, strong duality holds and $v^*(\hat{\mathbf{x}}, \hat{X}) = d^*(\hat{\mathbf{x}}, \hat{X})$. For any dual feasible solution $Z \in \mathcal{D}$,

$$Z \bullet \begin{pmatrix} 1 & \mathbf{x}^T \\ \mathbf{x} & X \end{pmatrix} \leq d^*(\mathbf{x}, X) = v^*(\mathbf{x}, X) = 0 \qquad \forall\ (\mathbf{x}, X) \in \mathcal{C}(\mathcal{F}_{SOC}).$$

That is, any dual solution $Z \in \mathcal{D}$ can be used to generate a valid inequality for $\mathcal{C}(\mathcal{F}_{SOC})$. Moreover, for any $(\hat{\mathbf{x}}, \hat{X}) \notin \mathcal{C}(\mathcal{F}_{SOC})$, let $\hat{Z}$ be an optimal solution to (2.13), then

$$\hat{Z} \bullet \begin{pmatrix} 1 & \hat{\mathbf{x}}^T \\ \hat{\mathbf{x}} & \hat{X} \end{pmatrix} = d^*(\hat{\mathbf{x}}, \hat{X}) = v^*(\hat{\mathbf{x}}, \hat{X}) > 0.$$

That is, $\left\{ (\mathbf{x}, X) \;\middle|\; \hat{Z} \bullet \begin{pmatrix} 1 & \mathbf{x}^T \\ \mathbf{x} & X \end{pmatrix} = 0 \right\}$ is a hyperplane strictly separating $(\hat{\mathbf{x}}, \hat{X})$ and $\mathcal{C}(\mathcal{F}_{SOC})$.

The last observation can be used to generate valid inequalities to tighten existing SDP relaxations of (2.3) in the $(\mathbf{x}, X)$-space. To show its effectiveness, we test the idea on the instances used in Sections 5.2 and 5.3 of [21]. Same as in [21], we consider two SDP relaxations with feasible regions $\mathcal{R}_{\text{shor}}$ and $\mathcal{R}_{\text{shor}} \cap \mathcal{R}_{\text{ksoc}}$, respectively, where

$$\mathcal{R}_{\text{shor}} := \{\, (\mathbf{x}, X) \mid \operatorname{tr}(X) \leq 1,\ \operatorname{tr}(X) \leq \mathbf{b}\mathbf{b}^T \bullet X - 2a\mathbf{b}^T\mathbf{x} + a^2,\ \mathbf{b}^T\mathbf{x} - a \geq 0,\ X \succeq \mathbf{x}\mathbf{x}^T \,\}$$

is the feasible region of the standard SDP (Shor) relaxation of (2.3), and $\mathcal{R}_{\text{ksoc}}$ is the set of $(\mathbf{x}, X)$ satisfying the linearized version of the Kronecker product constraint

$$\begin{pmatrix} 1 & \mathbf{x}^T \\ \mathbf{x} & I_n \end{pmatrix} \otimes \begin{pmatrix} \mathbf{b}^T\mathbf{x} - a & \mathbf{x}^T \\ \mathbf{x} & (\mathbf{b}^T\mathbf{x} - a)I_n \end{pmatrix} \succeq 0.$$

For more details about the Kronecker product constraints, we refer the readers to [4].

Let $(\hat{\mathbf{x}}, \hat{X})$ be an optimal solution to an SDP relaxation of (2.3), and let $\lambda_1 \geq \ldots \geq \lambda_{n+1}$ denote the eigenvalues of $\begin{pmatrix} 1 & \hat{\mathbf{x}}^T \\ \hat{\mathbf{x}} & \hat{X} \end{pmatrix}$. Following [21], we say that the SDP relaxation is exact if $\frac{\lambda_1}{\lambda_2} > 10^4$, i.e., the matrix $\begin{pmatrix} 1 & \hat{\mathbf{x}}^T \\ \hat{\mathbf{x}} & \hat{X} \end{pmatrix}$ is numerically rank-1. For each tested instance, we solve problem (2.6) with $\mathcal{C}(\mathcal{F})$ replaced with $\mathcal{R}_{\text{shor}}/\mathcal{R}_{\text{shor}} \cap \mathcal{R}_{\text{ksoc}}$ to get an initial optimal solution $(\hat{\mathbf{x}}, \hat{X})$. If the relaxation is inexact, we solve the separation problem (2.12) and generate a valid inequlity $\hat{Z} \bullet \begin{pmatrix} 1 & \mathbf{x}^T \\ \mathbf{x} & X \end{pmatrix} \leq 0$. We add the valid inequality to the relaxation and resolve. We repeat the process until the relaxation is exact.

We implement our experiments in Matlab 9.5 (R2018b) using CVX [27] to model the relaxations and MOSEK 9.1 [6] to solve them. We run the instances on an Intel(R) Core(TM) i7-8550U CPU @ 1.80GHz with four cores and 16GB of memory. For each dimension n, we test the 15,000 instances generated in [21], which can be found at `https://github.com/A-Eltved/strengthened_sdr`. We report the number of instances with an inexact initial ($\mathcal{R}_{\text{shor}}/\mathcal{R}_{\text{shor}} \cap \mathcal{R}_{\text{ksoc}}$) relaxation, the maximum and average number of cuts needed to close the gaps, and the percentage of instances whose gaps are closed with a single cut. The results are reported in Tables 2.1 and 2.2. (The numbers of instances with inexact initial relaxations are slightly different from [21], possibly due to different versions of platforms.)

For the Shor relaxation, we observe that the gaps of all instances are closed within 36 cuts. For the tighter relaxation with $\mathcal{R}_{\text{shor}} \cap \mathcal{R}_{\text{ksoc}}$, the gaps of all instances are closed

n	Inexact initial	Max cuts	Avg cuts	Closed with one cut (%)
2	7744	21	2.42	27.83
3	7634	36	1.99	50.21
4	7733	19	1.59	66.48
5	7704	16	1.40	76.43
6	7584	21	1.29	82.45
7	7648	10	1.22	86.36
8	7614	12	1.18	88.80
9	7566	7	1.13	91.05
10	7552	10	1.11	92.74

Table 2.1: Cut effectiveness for the $\mathcal{R}_{\text{shor}}$ relaxation of (2.3).

n	Inexact initial	Max cuts	Avg cuts	Closed with one cut (%)
2	15	3	1.40	73.33
3	50	4	1.64	54.00
4	36	4	1.69	52.78
5	27	6	1.52	62.96
6	15	3	1.40	66.67
7	13	3	1.31	76.92
8	12	2	1.08	91.67
9	5	1	1.00	100
10	5	2	1.20	80

Table 2.2: Cut effectiveness for the $\mathcal{R}_{\text{shor}} \cap \mathcal{R}_{\text{ksoc}}$ relaxation of (2.3).

within six cuts. On average, less than two cuts are needed for $n \geq 3$. A majority of the instances require only one cut to close the gaps. We remark here that with respect to time, each separation problem takes about 0.3 seconds to solve.

We conduct similar tests on the two-ball problem (2.2) and observe similar performance. Compared with (2.3), more cuts are needed to close the gaps on average, but most instances need no more than two cuts. The results are presented in Tables 2.3 and 2.4.

Acknowledgements This material is based upon Yang and Kelly's research supported by the Office of Naval Research under Award Number (N00014-20-1-2154) and Ouyang's research supported by the Office of Naval Research under Award Number (N00014-20-1-2089). We would like to thank the anonymous reviewers, whose suggestions help improve the completeness and readability of the paper. In particular, the proof of Lemma 1 is

n	Inexact initial	Max cuts	Avg cuts	Closed with one cut (%)
2	1403	19	2.45	7.98
3	1285	22	2.48	12.68
4	983	15	2.28	20.04
5	739	11	2.18	22.60
6	508	6	2.11	21.46
7	453	14	2.11	26.93
8	346	9	2.08	27.75
9	292	6	1.96	31.85
10	251	7	2.01	29.08

Table 2.3: Cut effectiveness for the $\mathcal{R}_{\text{shor}}$ relaxation of (2.2).

n	Inexact initial	Max cut	Avg cuts	Closed with one cut (%)
2	31	10	2.32	41.94
3	78	9	2.18	41.03
4	62	7	1.89	46.77
5	34	8	2.03	35.29
6	21	4	1.67	47.62
7	16	4	1.81	50.00
8	14	4	1.50	64.29
9	6	2	1.17	83.33
10	4	3	2.00	25.00

Table 2.4: Cut effectiveness for the $\mathcal{R}_{\text{shor}} \cap \mathcal{R}_{\text{ksoc}}$ relaxation of (2.2).

simplified based on a reviewer's suggestion.

Chapter 3

Semidefinite Programming Relaxation for Copositive Dual Pricing

In this article, we consider the unit commitment (UC) problem and the challenge its nonconvexity presents when it comes to pricing for the day-ahead electricity market. Currently, there is no consensus on which of the existing pricing mechanisms is best; however, there is a common desire for an improved pricing mechanism [34]. In practice, pricing mechanisms are based on shadow prices of a convex relaxation of the UC problem. Thus, motivated by works such as [29] and [22], we develop a semidefinite programming relaxation of the UC problem and examine its associated decomposition of the locational marginal price.

3.1 Introduction

In electricity markets, suppliers submit information about their generating capabilities and costs to an independent system operator (ISO). The ISO then needs to determine an optimal allocation based on predicted demand and set electricity prices [42]. To determine an optimal allocation, the ISO solves a unit commitment (UC) problem. Due to the nonconvexity of UC problems, it is often difficult to determine prices that not only cover the fuel costs of each generator, but also their fixed costs. This can lead to a "missing money" problem if chosen prices result in the revenue being less than the cost [56].

UC problems are often formulated as mixed-integer programs (MIPs) containing both continuous and binary variables [18]. While the MIP formulation is solvable even for very large-scale problems in a practically tractable amount of time, it is difficult to construct a "good" pricing mechanism without shadow prices. For this reason, Guo et al. [29] reformulated the MIP as a completely positive program (CPP) and then designed a pricing mechanism using the dual copositive program (COP). While Guo et al. show that their copositive duality pricing (CDP) has certain desirable properties, the CPP formulation may not be immediately implementable in practice due to the limitation of solvers. For this reason, we seek to relax the CPP formulation to a semidefinite programming (SDP) formulation and use its shadow prices to design a new pricing mechanism.

3.1.1 Literature Review

In this section, we review literature on pricing schemes for nonconvex energy market and convex relaxations of MIPs, including specifically UC problems.

Liberopoulos and Andrianesis [42] provides a review of several pricing schemes proposed for markets with nonconvexities. When evaluating a pricing mechanism, we recall that according to the Federal Energy Regulatory Commission (FERC) [34], pricing mechanisms should ideally be social welfare maximizing, transparent, incentive compatible, and ensure revenue adequacy.

Instead of solving the MIP formulation directly, a convex relaxation of the UC problem is often desired so that shadow prices may be obtained. Dropping the integrality constraints from UC yields a linear programming (LP) relaxation, but this LP relaxation tends to be weak. A better approach is the restricted pricing (RP) method formulated by O'Neill et al. [48]. Pricing in this method is based upon the linear program that results from fixing the integer variables at their optimal values, guaranteeing an optimal solution to the UC. While restricted pricing is commonly used in practice, it has some drawbacks such as large uplift payments (also known as make-whole payments). Another method known as convex hull pricing (CHP) [28, 32] has been shown to minimize uplift. Ruiz et al. [53] take a primal-dual approach that eliminates uplift through revenue adequacy constraints. Unfortunately, there is a tradeoff as their primal-dual approach does not necessarily support optimal UC decisions.

In search of a better pricing scheme, some works have begun exploring conic relaxations of the UC problem. Guo et al. [29] show that the UC problem can be reformulated as a completely-positive program. Others, such as Fattahi et al. [22] and Quarm and Madani [50], propose tight SDP relaxations for the UC problem. In the case of Quarm and Madani [50], the SDP relaxation contains smaller PSD conic constraint, which they show are decomposable by generator and time.

More generally, Bao et al. [9] show that the doubly nonnegative relaxation of QCQPs is equivalent to the SDP relaxation with first-level RLT constraints. Wang and Kılınc-Karzan [61] provide sufficient conditions for SDP relaxations of QCQPs to be tight, which could be used to provide sufficient conditions for tightness of SDP relaxations for UC problems. Kim and Kojima [40] and Bertsimas and Cory-Wright [10] propose obtaining tractable convex relaxations of non-convex problems by taking second-order cone (SOC) relaxations of their SDP relaxations.

In this work, we aim to develop a computationally tractable SDP relaxation of UC and construct an associated pricing scheme with desirable properties, such as low optimality gaps and low uplift costs.

3.2 CPP and SDP Reformulations of UC

Let $\mathcal{G}$ be a set of generators, $\mathcal{T}$ be a set of time slots, $\mathcal{N}$ be a set of nodes, and $\mathcal{L}$ be a set of lines. Energy prices are based on the following UC problem. Similar to the UC formulation in [22], we use a shift factor formulation for transmission constraints.

$$\min \quad \sum_{g\in\mathcal{G}}\sum_{t\in\mathcal{T}}\left(f_g(p_{gt}) + c_g^u u_{gt} + c_g^z z_{gt}\right) \tag{3.1a}$$

$$\text{s.t.} \quad \sum_{g\in\mathcal{G}} p_{gt} = \sum_{k\in\mathcal{N}} d_{kt} \qquad \forall t\in\mathcal{T} \qquad (\lambda_t) \tag{3.1b}$$

$$B_{ij}\sum_{k=1}^{|\mathcal{N}|}\left((s_{ik}-s_{jk})\,(\sum_{g\in\mathcal{G}_k} p_{gt} - d_{kt})\right) \geq p_{ij}^{\text{Trans,min}} \qquad \forall (i,j)\in\mathcal{L}, t\in\mathcal{T} \qquad (\xi_{ijt}^{\min}) \tag{3.1c}$$

$$B_{ij}\sum_{k=1}^{|\mathcal{N}|}\left((s_{ik}-s_{jk})\,(\sum_{g\in\mathcal{G}_k} p_{gt} - d_{kt})\right) \leq p_{ij}^{\text{Trans,max}} \qquad \forall (i,j)\in\mathcal{L}, t\in\mathcal{T} \qquad (\xi_{ijt}^{\max}) \tag{3.1d}$$

$$p_{gt} \geq p_g^{\min} z_{gt} \qquad \forall g\in\mathcal{G}, t\in\mathcal{T} \qquad (\sigma_{gt}^{\min}) \tag{3.1e}$$

$$p_{gt} \leq p_g^{\max} z_{gt} \qquad \forall g\in\mathcal{G}, t\in\mathcal{T} \qquad (\sigma_{gt}^{\max}) \tag{3.1f}$$

$$u_{gt} - v_{gt} = z_{gt} - z_{g,t-1} \qquad \forall g\in\mathcal{G}, t\in\mathcal{T}\setminus\{1\} \ (\mu_{gt}) \tag{3.1g}$$

$$p_{gt} - p_{g,t-1} \leq r_g^{\text{U}} z_{g,t-1} + p_g^{\min} u_{gt} \qquad \forall g\in\mathcal{G}, t\in\mathcal{T}\setminus\{1\} \ (\beta_{gt}^U) \tag{3.1h}$$

$$p_{g,t-1} - p_{gt} \leq r_g^{\text{D}} z_{gt} + p_g^{\min} v_{gt} \qquad \forall g\in\mathcal{G}, t\in\mathcal{T}\setminus\{1\} \ (\beta_{gt}^D) \tag{3.1i}$$

$$\sum_{\tau=\max(1,t-\Delta_g^{\text{Up}})}^{t} u_{g\tau} \leq z_{gt} \qquad \forall g\in\mathcal{G}, t\in\mathcal{T}\setminus\{1\} \ (\gamma_{gt}^U) \tag{3.1j}$$

$$\sum_{\tau=\max(1,t-\Delta_g^{\text{Down}})}^{t} v_{g\tau} \leq 1 - z_{gt} \qquad \forall g\in\mathcal{G}, t\in\mathcal{T}\setminus\{1\} \ (\gamma_{gt}^D) \tag{3.1k}$$

$$u_{gt}, v_{gt} \leq 1 \qquad \forall g\in\mathcal{G}, t\in\mathcal{T} \qquad (\phi_{gt}^u, \phi_{gt}^v) \tag{3.1l}$$

$$z_{gt} \in \{0,1\} \qquad \forall g\in\mathcal{G}, t\in\mathcal{T} \qquad (\phi_{gt}^z) \tag{3.1m}$$

$$p_{gt}, u_{gt}, v_{gt}, z_{gt} \geq 0 \qquad \forall g\in\mathcal{G}, t\in\mathcal{T} \tag{3.1n}$$

where B is the susceptance matrix and s_{ij} is the $(i,j)th$ entry of the shift factor matrix. When computing the shift factor matrix, we set the bus $i=1$ as the reference bus. As for the variables, p_{gt} denotes the units of electricity produced by generator g during time slot t, $u_{gt}=1$ if and only if generator g is started up during time slot t, $v_{gt}=1$ if and only if generator g is shutdown during time slot t, and $z_{gt}=1$ if and only if generator g is on during time slot t. Note that z_{gt}, u_{gt}, and v_{gt} are all binary variables, but we do not explicitly impose binary constraints on u_{gt} and v_{gt} because these binary constraints are implied by constraints (3.1g), (3.1l), (3.1m), and (3.1n). The Greek letters in the rightmost column denote the corresponding dual variables for the linear programming relaxation of (3.1).

The objective (3.1a) is the total cost, which includes production cost, fixed startup cost, and fixed cost of having a generator on. Constraints (3.1b) ensure that the total demand is satisfied at each hour. In constraints (3.1c) and (3.1d), the left-hand side represents the flow of the network using a shift factor formulation, and the right-hand side imposes bounds based on line limitations. Constraints (3.1e) and (3.1f) bound production levels. Constraints (3.1g) ensure $u_{gt}=1$ only when a generator gets turned on and $v_{gt}=1$ only when a generator gets turned off. Constraints (3.1h) and (3.1i) are ramping up and ramping down constraints, respectively. Constraints (3.1j) and (3.1k) are minimum up time and minimum down time constraints, respectively.

We transform the MIP to a CPP. To simplify the transformation, the MIP above can be rewritten to the following problem $\mathcal{UC}$:

$$
\begin{aligned}
\mathcal{UC}: \quad \min \quad & f(\mathbf{x}) && (3.2a)\\
\text{s.t.} \quad & \mathbf{a}_j^\top \mathbf{x} = b_j \quad \forall j = 1, ..., m && (3.2b)\\
& \mathbf{x} \in \mathbb{R}_+^n && (3.2c)\\
& x_i \in \{0,1\} \quad \forall i \in \mathcal{B}. && (3.2d)
\end{aligned}
$$

where $\mathbf{x}$ contains the original variables, as well as slack variables. $f(\mathbf{x})$ denotes the objective function (3.1a), and $\mathcal{B}$ denotes the set of indices corresponding to the binary variables z_{gt}.

Constraints (3.2b) include constraints (3.1b) - (3.1l), as well as the upper bound for the binary variable z_{gt}, after adding slack variables.

As shown by [29], we can use the fact that mixed-binary quadratic programs can be reformulated as completely positive programs [13] to reformulate $\mathcal{UC}$ to the following CPP:

$$
\begin{aligned}
\mathcal{UC}^{\mathrm{CPP}}: \quad \min \quad & f(\mathbf{x}) && && (3.3a)\\
\text{s.t.} \quad & \mathbf{a}_j^\top \mathbf{x} = b_j && \forall j = 1, ..., m && (3.3b)\\
& \mathrm{Tr}(\mathbf{a}_j \mathbf{a}_j^\top X) = b_j^2 && \forall j = 1, ..., m && (3.3c)\\
& x_i = X_{ii} && \forall i \in \mathcal{B} && (3.3d)\\
& Y \in \mathcal{C}_{n+1}^*. && && (3.3e)
\end{aligned}
$$

where $Y = \begin{bmatrix} 1 & \mathbf{x}^\top \\ \mathbf{x} & X \end{bmatrix}$. Constraints (3.3b) are the same as (3.2b). Constraints (3.3c) are first-level RLT constraints, and constraints (3.3d) are RLT constraints for each binary variable z_{gt}. In constraint (3.3d), $\mathcal{C}_{n+1}^*$ denotes the $(n+1)$-dimensional completely positive cone.

It was shown in [29] that shadow prices of $\mathcal{UC}^{\mathrm{CPP}}$ can be used to define electricity prices that make the system operator and generators budget-balanced, support optimal unit commitment decisions, and under certain conditions, are individually rational.

However, to the best of our knowledge, no practical solvers are capable of solving CPPs exactly. Indeed, to our knowledge, the most scalable exact method for CPPs is Cylindrical Algebraic Decomposition [8,52], which is a universal quantifier elimination technique that solves CPPs in doubly exponential time, and cannot scale to successfully solve real problem instances. Accordingly, in the literature, CPPs are usually solved approximately via SDP relaxations, which require that Y is doubly nonnegative (DNN), by replacing the completely positive conic constraint (3.3e) with constraints (3.4) below.

$$Y \in \mathcal{S}^{+}_{n+1} \tag{3.4a}$$

$$Y \geq 0. \tag{3.4b}$$

We denote the DNN relaxation of $\mathcal{UC}^{\text{CPP}}$ as $\mathcal{UC}^{\text{SDP}} := \min\{f(\mathbf{x}) | (3.3\text{b}) - (3.3\text{d}), (3.4)\}$.

3.3 Pricing with SDP Relaxations

The price at node i equals the marginal total cost increase when demand at node i increases. Consider the value function $v(\mathbf{d})$ of $\mathcal{UC}^{\text{SDP}}$. If there is a unique subgradient at d_{it}, then the subgradient provides the marginal increase in total cost. The subgradient equals the partial derivative of $v(\mathbf{d})$ at d_{kt}:

$$\begin{aligned} \lambda_t + 2\hat{\lambda}_t \sum_{k' \in \mathcal{N}} d_{k't} &+ \sum_{(i,j) \in \mathcal{L}} B_{ij}(s_{ik} - s_{jk})(\xi^{\min}_{ijt} + \xi^{\max}_{ijt}) \\ &+ 2 \sum_{(i,j) \in \mathcal{L}} B^2_{ij}(s_{ik} - s_{jk}) \Big(\sum_{k' \in \mathcal{N}} (s_{ik'} - s_{jk'}) d_{k't} \Big) (\hat{\xi}^{\min}_{ijt} + \hat{\xi}^{\max}_{ijt}) \\ &+ 2 \sum_{(i,j) \in \mathcal{L}} B_{ij}(s_{ik} - s_{jk})(p^{\text{Trans,min}}_{ij} \hat{\xi}^{\min}_{ijt} + p^{\text{Trans,max}}_{ij} \hat{\xi}^{\max}_{ijt}) \end{aligned} \tag{3.5}$$

where $\hat{\lambda}_t$ is the dual variable associated with the RLT constraint that is obtained by squaring constraint (3.1b). Similarly, $\hat{\xi}^{\min}_{i'j't}$ and $\hat{\xi}^{\max}_{i'j't}$ are the dual variables associated with the RLT constraints that are obtained by squaring constraints (3.1c) and (3.1d), respectively.

Expression (3.5) is a linear function of $\mathbf{d}$. In what follows, we focus on the decomposition and interpretation of λ_t, which is commonly used as the locational marginal prices (LMPs) at the reference bus.

3.3.1 SDP Relaxation with $X \in \mathcal{S}_n^+$ and LMP Decomposition

To obtain a decomposition of the LMP at the reference bus (i.e., λ_t), we consider the dual constraint for p_{gt} with g locating at the reference bus:

$$\begin{aligned}\lambda_t = f'(p_{gt}) - \sum_{(i,j)\in\mathcal{L}} B_{ij}(s_{ik} - s_{jk})(\xi_{ijt}^{\min} + \xi_{ijt}^{\max}) - \sigma_{gt}^{\min} - \sigma_{gt}^{\max} \\ - \beta_{gt}^{U}\mathbb{1}_{t\neq 1} + \beta_{g,t+1}^{U}\mathbb{1}_{t\neq|\mathcal{T}|} - \beta_{g,t+1}^{D}\mathbb{1}_{t\neq|\mathcal{T}|} + \beta_{gt}^{D}\mathbb{1}_{t\neq 1} - \eta_{gt}^{p} - \Omega_{gt}^{p}\end{aligned} \tag{3.6}$$

where $\mathbb{1}$ denotes the indicator function, k is the node in which generator g locates, η_{gt}^p is the dual variable associated with the nonnegativity constraint of p_{gt}. The notation Ω is the dual variable associated with the PSD constraint (3.4a), and we use Ω_{gt}^p to denote the entry of Ω that corresponds to variable p_{gt}.

All variables except Ω_{gt}^p in this constraint have physical meanings. For this reason, the LMP decomposition would be more interpretable if it did not contain Ω_{gt}^p. Thus, to obtain a more interpretable LMP decomposition, we can replace $Y \in \mathcal{S}_{n+1}^+$ with a lower dimensional positive semidefinite (PSD) constraint and a set of RLT constraints. In the following theorem, we show that this can be done without compromising the tightness of the formulation. Note that constraint (3.7a) can be obtained by multiplying the constraint $\mathbf{a}_k^\top \mathbf{x} = b_k$ with all variables in $\mathbf{x}$.

Theorem 11. *Let $\mathbf{a}_k^\top \mathbf{x} = b_k$ be a constraint in* (3.3b) *with $b_k \neq 0$, then for $\mathcal{UC}^{SDP}$, the PSD conic constraint* (3.4a) *can be equivalently replaced by the following RLT and conic constraints:*

$$X\mathbf{a}_k = b_k\mathbf{x} \tag{3.7a}$$

$$X \in \mathcal{S}_n^+ \tag{3.7b}$$

Proof. Let $\mathbf{a}_k^T\mathbf{x} = b_k$ be a constraint in (3.3b) with $b_k \neq 0$. Also, let $\mathbf{y} := \frac{1}{b_k}\mathbf{e}_k$ and $\boldsymbol{\alpha} := A^T\mathbf{y}$. Then,

$$\boldsymbol{\alpha}^T\mathbf{x} = \mathbf{y}^T A\mathbf{x} = \mathbf{y}^T\mathbf{b} = \frac{1}{b_k}b_k = 1$$

for all $\mathbf{x}$ feasible to (3.2). Therefore, $X\boldsymbol{\alpha} = \mathbf{x}$ is a valid inequality for the SDP relaxation of (3.2). Thus, with the addition of the constraint $X\boldsymbol{\alpha} = \mathbf{x}$, the original PSD constraint can be simplified from

$$\begin{pmatrix} 1 & \boldsymbol{\alpha}^T X \\ X\boldsymbol{\alpha} & X \end{pmatrix} \succeq 0$$

to $X \succeq 0$ because

$$\begin{pmatrix} 1 & \boldsymbol{\alpha}^T X \\ X\boldsymbol{\alpha} & X \end{pmatrix} = \begin{pmatrix} \boldsymbol{\alpha} & I \end{pmatrix}^T X \begin{pmatrix} \boldsymbol{\alpha} & I \end{pmatrix}.$$

Now, note that since that $\boldsymbol{\alpha} = A^T\mathbf{y} = \frac{1}{b_k}\mathbf{a}_k$, $X\boldsymbol{\alpha} = \mathbf{x}$ is equivalent to $X\mathbf{a}_k = b_k\mathbf{x}$. Hence, the PSD conic constraint (3.4a) can be replaced by constraints (3.7a) and (3.7b).

For the other direction, it is clear that $Y \succeq 0$ implies $X \succeq 0$. Furthermore, in Proposition 8.3 of [14], it is shown that if $Y \succeq 0$, then constraints (3.3b) and (3.3c) are equivalent to constraints (3.3b) and $AX = \mathbf{b}\mathbf{x}^T$. □

To make the LMP decomposition interpretable, we drop the constraints in (3.7a) that correspond to p_{gt}'s, so the LMP decomposition does not contain dual variables of (3.7a). As shown in the numerical experiments of Section 3.4.2, dropping those constraints do not impact the tightness too much.

Due to the potential complications introduced by an asymmetric coefficient matrix in conic program, which can lead to the dual problem including asymmetric feasible matrices [29], we rewrite (3.7a) (without constraints containing $\mathbf{p}$) to ensure its coefficient matrix is symmetric:

$$b_k x_l = \mathrm{Tr}\left(\frac{\mathbf{a}_k\mathbf{e}_l^\top + \mathbf{e}_l\mathbf{a}_k^\top}{2}X\right), \forall l \in \hat{\mathcal{I}} \tag{3.8}$$

where $\hat{\mathcal{I}}$ is a subset of variable indices without indices for $\mathbf{p}$.

In sum, we obtain LMP from the SDP relaxation $\mathcal{UC}^{\mathrm{SDP}'} := \min\{f(\mathbf{x})|(3.3\mathrm{b}) - (3.3\mathrm{d}), (3.4\mathrm{b}), (3.7\mathrm{b}), (3.8)\}$, with the following LMP decomposition for the price at the reference bus ($\forall t \in \mathcal{T}$):

$$\begin{aligned}\lambda_t = f'(p_{gt}) - \sum_{(i,j)\in\mathcal{L}} B_{ij}(s_{ik} - s_{jk})(\xi_{ijt}^{\min} + \xi_{ijt}^{\max}) - \sigma_{gt}^{\min} - \sigma_{gt}^{\max} \\ - \beta_{gt}^{U}\mathbb{1}_{t\neq 1} + \beta_{g,t+1}^{U}\mathbb{1}_{t\neq|\mathcal{T}|} - \beta_{g,t+1}^{D}\mathbb{1}_{t\neq|\mathcal{T}|} + \beta_{gt}^{D}\mathbb{1}_{t\neq 1} - \eta_{gt}^{p}\end{aligned} \tag{3.9}$$

For a particular node $k \in \mathcal{N}$ at time $t \in \mathcal{T}$, the LMP can be obtained using the following formulation, which is the constant part of (3.5) without the terms containing $\hat{\xi}_{ijt}^{\min}$ and $\hat{\xi}_{ijt}^{\max}$:

$$\lambda_{kt} = \lambda_t + \sum_{(i,j)\in\mathcal{L}} B_{ij}(s_{ik} - s_{jk})(\xi_{ijt}^{\min} + \xi_{ijt}^{\max}).$$

3.3.2 An Alternative SDP Relaxation and Pricing

In this section, we consider the SDP reformulation for UC problem proposed in [22]. We prove that their formulation, which we denote as $\mathcal{UC}_2^{\text{SDP}}$, is equivalent to $\mathcal{UC}^{\text{SDP}}$ in terms of tightness. However, $\mathcal{UC}_2^{\text{SDP}}$ could lead to less interpretable prices.

The SDP reformulation from [22] can be written as follows:

$$\mathcal{UC}_2^{\text{SDP}}: \quad \min \quad f(\tilde{\mathbf{x}}) \tag{3.10a}$$

$$\text{s.t.} \quad \tilde{\mathbf{a}}_j^{\top}\tilde{\mathbf{x}} \geq \tilde{b}_j \qquad \forall j = 1, ..., \tilde{m} \tag{3.10b}$$

$$\tilde{\mathbf{a}}_j^{\top}\tilde{X} \geq \tilde{b}_j\tilde{\mathbf{x}}^{\top} \qquad \forall j = 1, ..., \tilde{m} \tag{3.10c}$$

$$\tilde{\mathbf{a}}_i^{T}\tilde{X}\tilde{\mathbf{a}}_j - \tilde{b}_i\tilde{\mathbf{x}}^{T}\tilde{\mathbf{a}}_j - \tilde{\mathbf{a}}_i^{T}\tilde{\mathbf{x}}\tilde{b}_j^{\top} + \tilde{b}_i\tilde{b}_j \geq 0 \quad \forall i, j = 1, ..., \tilde{m} \tag{3.10d}$$

$$\tilde{x}_i = \tilde{X}_{ii} \qquad \forall i \in \mathcal{B} \tag{3.10e}$$

$$\tilde{Y} \in \mathcal{S}_{\tilde{n}+1}^{+} \tag{3.10f}$$

$$\tilde{Y} \geq 0, \tag{3.10g}$$

where $\tilde{\mathbf{x}} \in \mathbb{R}^{\tilde{n}}$ is a vector of all variables in $\mathcal{UC}$. Constraints (3.10b) include constraints (3.1b) - (3.1l) and upper bounds for binary variables, with equality constraints equivalently converted to two inequality constraints.

In Theorem 12, we show that SDP relaxations $\mathcal{UC}^{\mathrm{SDP}}$ and $\mathcal{UC}_2^{\mathrm{SDP}}$ have the same strength:

Theorem 12. *$opt(\mathcal{UC}^{SDP}) = opt(\mathcal{UC}_2^{SDP})$*

Proof. To begin, consider two linear systems of inequalities:

$$A\mathbf{x} = \mathbf{b},\ \mathbf{x} \geq \mathbf{0}, \tag{3.11}$$

$$\tilde{A}\tilde{\mathbf{x}} \geq \tilde{\mathbf{b}},\ \tilde{\mathbf{x}} \geq \mathbf{0}, \tag{3.12}$$

where $\tilde{A}$ is $m \times n$, $A = [\tilde{A} \quad -I_m]$ and $\mathbf{b} = \tilde{\mathbf{b}}$. The standard PSD+RLT relaxation of (3.11) is as follows:

$$A\mathbf{x} = \mathbf{b} \tag{3.13a}$$

$$AX = \mathbf{b}\mathbf{x}^T \tag{3.13b}$$

$$\mathbf{x} \geq \mathbf{0} \tag{3.13c}$$

$$X \geq 0 \tag{3.13d}$$

$$\begin{pmatrix} 1 & \mathbf{x}^T \\ \mathbf{x} & X \end{pmatrix} \succeq 0. \tag{3.13e}$$

Similarly, the standard PSD+RLT relaxation of (3.12) is given by:

$$\tilde{A}\tilde{\mathbf{x}} \geq \tilde{\mathbf{b}} \tag{3.14a}$$

$$\tilde{A}\tilde{X} \geq \tilde{\mathbf{b}}\tilde{\mathbf{x}}^T \tag{3.14b}$$

$$\tilde{A}\tilde{X}\tilde{A}^T - \tilde{\mathbf{b}}\tilde{\mathbf{x}}^T\tilde{A}^T - \tilde{A}\tilde{\mathbf{x}}\mathbf{b}^T + \tilde{\mathbf{b}}\tilde{\mathbf{b}}^T \geq 0 \tag{3.14c}$$

$$\tilde{\mathbf{x}} \geq \mathbf{0} \tag{3.14d}$$

$$\tilde{X} \geq 0 \tag{3.14e}$$

$$\begin{pmatrix} 1 & \tilde{\mathbf{x}}^T \\ \tilde{\mathbf{x}} & \tilde{X} \end{pmatrix} \succeq 0. \tag{3.14f}$$

We will now show that a point $(\tilde{\mathbf{x}}, \tilde{X})$ satisfies (3.14) if and only if there exist $\mathbf{s}, Y,$ and Z such that $\mathbf{x} = (\tilde{\mathbf{x}}, \mathbf{s})$ and $X = \begin{pmatrix} \tilde{X} & Y^T \\ Y & Z \end{pmatrix}$ satisfies (3.13). Define

$$\mathbf{s} = \tilde{A}\tilde{\mathbf{x}} - \tilde{\mathbf{b}} \tag{3.15a}$$

$$Y = \tilde{A}\tilde{X} - \tilde{\mathbf{b}}\tilde{\mathbf{x}}^T \tag{3.15b}$$

$$Z = \tilde{A}Y^T - \tilde{\mathbf{b}}\mathbf{s}^T = \tilde{A}\tilde{X}\tilde{A}^T - \tilde{A}\tilde{\mathbf{x}}\tilde{\mathbf{b}}^T - \tilde{\mathbf{b}}\tilde{\mathbf{x}}^T\tilde{A}^T + \tilde{\mathbf{b}}\tilde{\mathbf{b}}^T \tag{3.15c}$$

Then, for $\mathbf{x} = (\tilde{\mathbf{x}}, \mathbf{s})$ and $X = \begin{pmatrix} \tilde{X} & Y^T \\ Y & Z \end{pmatrix}$, constraints (3.13a) – (3.13d) are satisfied. For (3.13e), note that

$$\begin{pmatrix} 1 & \mathbf{x}^T \\ \mathbf{x} & X \end{pmatrix} = \begin{pmatrix} 1 & \tilde{\mathbf{x}}^T & \mathbf{s}^T \\ \tilde{\mathbf{x}} & \tilde{X} & Y^T \\ \mathbf{s} & Y & Z \end{pmatrix} = \begin{pmatrix} 1 & \mathbf{0}^T \\ \mathbf{0} & I_n \\ -\tilde{\mathbf{b}} & \tilde{A} \end{pmatrix} \begin{pmatrix} 1 & \tilde{\mathbf{x}}^T \\ \tilde{\mathbf{x}} & \tilde{X} \end{pmatrix} \begin{pmatrix} 1 & \mathbf{0}^T \\ \mathbf{0} & I_n \\ -\tilde{\mathbf{b}} & \tilde{A} \end{pmatrix}^T \succeq 0.$$

("$\Leftarrow$") Suppose that $\mathbf{x} = (\tilde{\mathbf{x}}, \mathbf{s})$ and $X = \begin{pmatrix} \tilde{X} & Y^T \\ Y & Z \end{pmatrix}$ is a solution to (3.13). The proof is straightforward noting that (3.13a) and (3.13b) imply (3.15).

Finally, note that with the presence of (3.13a) and (3.13e), (3.13b) is equivalent to $AXA^T = \mathbf{b}\mathbf{b}^T$ which is further equivalent to $\operatorname{diag}\left(AXA^T\right) = \mathbf{b} \circ \mathbf{b}$, where $\circ$ represents the Hadamard product. Hence, if we ignore the constraint $x_{ii} = X_{ii}$ ($\tilde{x}_{ii} = \tilde{X}_{ii}$) for all $i \in \mathcal{B}$ that is in both $\mathcal{UC}^{\text{SDP}}$ and $\mathcal{UC}_2^{\text{SDP}}$, then the form of $\mathcal{UC}^{\text{SDP}}$ is equivalent to (3.13) and $\mathcal{UC}_2^{\text{SDP}}$ has the same form as (3.14). Thus, without the aforementioned constraint, they are equivalent. This equivalency remains when we add back in the constraints $x_{ii} = X_{ii}$ ($\tilde{x}_{ii} = \tilde{X}_{ii}$) for all $i \in \mathcal{B}$ corresponding to the original binary variables z_{gt}. □

Compared with $\mathcal{UC}^{\text{SDP}}$, the conic constraint of $\mathcal{UC}_2^{\text{SDP}}$ has lower dimension, as $\tilde{\mathbf{x}}$ does not contain slack variables. Therefore,$\mathcal{UC}_2^{\text{SDP}}$ could be more tractable. On the other hand, because the RLT constraints (3.10d) include variables $\tilde{\mathbf{x}}$, this adds additional terms to the constant components of $v(\mathbf{d})$, leading to a less interpretable LMP decomposition.

3.4 Numerical Experiments

In our numerical experiments, the programming language we use is Julia 1.6.7. The optimization problems are modeled in JuMP 1.13 [41] and solved with Mosek 10.1.3 [7].

When the size of a matrix is large, it can take a long time to build the optimization models in Julia via matrix multiplication. To reduce the building time when constructing the expression of the objective or the left-hand side of a constraint, we first ensure the coefficient of each variable is nonzero before multiplying it with the variable, since comparison takes less CPU time than multiplication [30]. This saves much time as the A matrix is relatively sparse.

3.4.1 A Simple UC Problem

Consider the following UC problem which has a loose LP relaxation. In this simple example, there are two generators, and we are solving over one time period without considering startup costs.

$$\begin{aligned}
\min \quad & p_1 + 2p_2 + 3z_1 + z_2 && \text{(3.16a)} \\
\text{s.t.} \quad & p_1 + p_2 = 0.65 && \text{(3.16b)} \\
& p_1 \leq 0.45 z_1 && \text{(3.16c)} \\
& p_2 \leq 0.4 z_2 && \text{(3.16d)} \\
& p_1 \geq 0.4 z_1 && \text{(3.16e)} \\
& p_2 \geq 0.15 z_2 && \text{(3.16f)} \\
& z_1, z_2 \in \{0, 1\} && \text{(3.16g)}
\end{aligned}$$

The optimal solution is $z_1 = z_2 = 1$ with an optimal objective value of 4.85. For the LP relaxation, $z_1 = 0.56, z_2 = 1$, with an optimal objective value of 3.72. In Table 3.1, we compare the LMP at the reference bus from the RP relaxation and from various SDP relaxations discussed in previous sections. All of the formulations in Table 3.1 have an optimality gap of less than 0.01%.

Method	λ_t	Uplift
RP	2	Y
$\mathcal{UC}^{\text{SDP}}$	214.44	N
$\mathcal{UC}_2^{\text{SDP}}$	67.26	N
$\mathcal{UC}^{\text{SDP}'}$	57.88	N

Table 3.1: Comparison of the LMP at the reference bus

In the optimal solution of (3.16), p_1 is at its upper bound of 0.45, while $p_2 = 0.2$. Since neither of the bounds on p_2 are active at the optimal solution, λ_t is equal to the marginal cost of generator 2 as expected for the RP method. Because this does not account for the fixed cost of having the generators on, it does not generate enough revenue to cover the total costs, and an uplift payment is required to cover the difference. Looking at the remaining three methods, we see that λ_t is considerably larger for the SDP methods, leading to 0 uplift payment.

3.4.2 Tightness of SDP for IEEE 14-Bus System

In this section, we compare the optimality gap of $\mathcal{UC}^{\text{SDP}}$, $\mathcal{UC}^{\text{SDP}'}$, and that of the standard LP relaxation on IEEE 14-bus instance. Figure 3.1 shows the optimality gap for 18 load scenarios for the IEEE 14-bus system with 5 generators over one hour. The load factors used are 0.1i for $i = 1, \ldots, 18$. Similarly, Figure 3.2 shows the optimality gap for 17 line rating scenarios for the IEEE 14-bus system with 5 generators over one time hour. The line ratings used are 15 + 5i for $i = 1, \ldots, 17$.

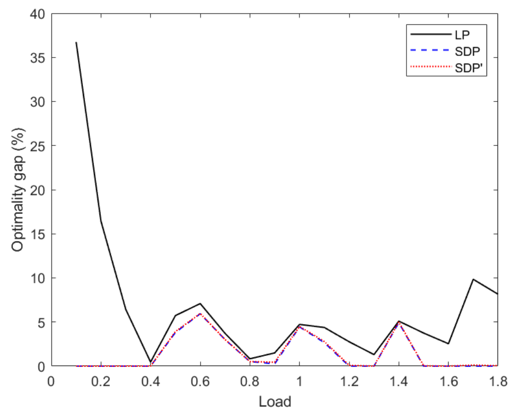


Figure 3.1: 18 load scenarios for the IEEE 14-bus system with 5 generators over one time slot with a line rating of 50.

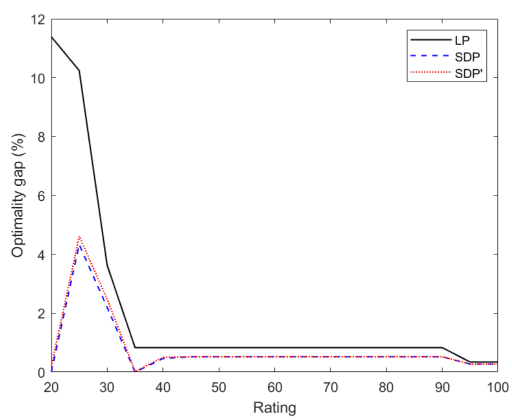


Figure 3.2: 17 line rating scenarios for the IEEE 14-bus system with 5 generators over one time slot with a load factor of 0.8.

In Figures 3.1 and 3.2, we observe that both $\mathcal{UC}^{\text{SDP}}$ and $\mathcal{UC}^{\text{SDP}'}$ are tighter than the standard LP relaxation. As the LP relaxations is already quite tight for some of these scenarios, there is not always a large difference between the three optimality gaps; however, the SDP relaxations are much tighter when the load or line rating is low. We also observe that even though we remove the constraints of (3.7a) corresponding to p_{gt}, $\mathcal{UC}^{\text{SDP}'}$ is often still as tight as $\mathcal{UC}^{\text{SDP}}$.

3.5 Conclusion

We obtain an SDP relaxation of the UC problem from its exact CPP reformulation. We use this relaxation to obtain the LMPs, and show that those prices are more interpretable compared with LMPs from an alternative SDP relaxation in the literature. Our numerical experiments show that our SDP relaxation is tight. While there is still room for improvement, it also shows promise in producing LMPs that reduce uplift costs.

Chapter 4

Affine Hull Algorithms

While algorithms for computing the affine hull of a polyhedron exist, these algorithms can be inefficient when the polyhedron has a large number of implicit equalities. Using strong duality and an LP method for finding a maximal element of a non-negative convex set, we derive a new algorithm which computes the affine hull of a polyhedron in a single iteration. We include computational tests which show that our algorithm performs well on benchmark instances and is quicker than Fukuda's algorithm.

4.1 Introduction

Efficient algorithms for finding the implicit equalities of a polyhedron are of interest due to the relationship between a polyhedron's implicit equalities and its affine hull and the relationship between the affine hull of a feasible region and corresponding SDP relaxations. In [58], Tunçel notes that SDP relaxations are popular for a variety of nonconvex optimization problems (e.g., combinatorial optimization problems). Tunçel also points out that SDPs are typically solved via an interior-point method, and he observes that many of these interior-point algorithms are more stable when Slater's condition holds. Tunçel proves that by determining the affine hull of a feasible set, we can detemine an affine space in the lifted space such that Slater's condition is guaranteed [58].

The practical implementation of finding an affine set containing a mixed-binary feasible region for facial reduction (FR) is proposed in [33]. The authors note that if a feasible region is defined by a polyhedron and binary constraints, then the affine hull of the feasible region is a subset of the affine hull of the polyhedron. Thus, the first step of their affine FR algorithm is to determine the affine hull of the polyhdron. Because facial reduction is a preprocessing step, this step and the facial reduction algorithm as a whole should be quick and effective [33].

In this paper, we examine, compare, and seek to improve algorithms for finding implicit equalities among the inequalities defining a polyhedron. The algorithms that we look at are based on one of two ideas. One idea is to add slack variables to the inequality constraints, and then search for a solution that maximizes the number of nonzero slack variables. The other idea is grounded in strict complimentary slackness.

4.1.1 Literature Review

While there are already algorithms for computing the affine hull of a polyhedron, the performance of these algorithms is to be improved. For example, the algorithm described in [24], which we will refer to as Fukuda's algorithm, is an iterative method, and in worst case, it finds only one implicit equality per iteration. The same holds for the algorithm in Corollary 14.lf. of [55]. In [23], the authors give a few LPs that may be used to find implicit inequalities; however, we note that there may be numerical difficulties when a method depends on whether or not a number is strictly positive. In [45], Mehdiloo explains how the characteristic cone of a polyhedron can be used to construct an LP that will find a relative interior point.

4.1.2 Preliminaries

Let $P = \{\mathbf{x} \in \mathbb{R}^n \mid A\mathbf{x} \leq \mathbf{b}\}$ be a polyhedron, where $A \in \mathbb{R}^{m\times n}$ and $\mathbf{b} \in \mathbb{R}^m$. For any $I \subseteq \{1, \ldots, m\}$, we let $A_I\mathbf{x} \leq \mathbf{b}_I$ denote the set of linear inequalities in $A\mathbf{x} \leq \mathbf{b}$ associated with I. Similarly, $\mathbf{a}_i^T\mathbf{x} \leq b_i$ denotes the ith inequality in the linear system

$A\mathbf{x} \leq \mathbf{b}$. We call an inequality $\mathbf{a}_i^T\mathbf{x} \leq b_i$ an *implicit equality* if $\mathbf{a}_i^T\mathbf{x} = b_i$ for every $\mathbf{x}$ in the polyhedron. Denote by $A^=\mathbf{x} \leq \mathbf{b}^=$ the system comprising all implicit equalities in $A\mathbf{x} \leq \mathbf{b}$. While the *affine hull* of P is defined by

$$\text{aff}(P) := \left\{ \sum_{i=1}^{k} \lambda_i \mathbf{x}_i \;\middle|\; \mathbf{x}_i \in P \text{ for all } i \text{ and } \sum_{i=1}^{k} \lambda_i = 1 \text{ for some } k \right\},$$

it is well-known that the affine hull of a polyhedron satisfies $\text{aff}(P) = \{\mathbf{x} \in \mathbb{R}^n \mid A^=\mathbf{x} = \mathbf{b}^=\}$. (Note that if P is empty, then it is vacuously true that all inequalities are implicit equalities and thus $\text{aff}(P) = \{\mathbf{x} \in \mathbb{R}^n \mid A\mathbf{x} = \mathbf{b}\} = \emptyset$.) We now give a simple example of a polyhedron that contains an implicit equality.

Example 1. *Let*

$$P = \{\mathbf{x} \in \mathbb{R}^3 \mid 4x_1 - x_3 \leq 0,\ -3x_2 - 2x_3 \leq -6,\ x_2 \leq 2,\ 2x_3 \leq 0\}.$$

It can be shown that for all $\mathbf{x} \in P$, $x_2 = 2$ and $x_3 = 0$. Thus, the second, third, and fourth constraints defining P are implicit equalities as they are always active. This means that P can equivalently be written as $\{\mathbf{x} \in \mathbb{R}^3 \mid 4x_1 - x_3 \leq 0,\ -3x_2 - 2x_3 = -6,\ x_2 = 2,\ 2x_3 = 0\}$.

4.2 Revisit of the existing methods

4.2.1 Fukuda's algorithm

In [24], Fukuda introduces an algorithm to compute the dimension of a polyhedron. The algorithm solves a linear program in each iteration, and it essentially finds the affine hull of the polyhedron. In this section, we revisit the algorithm and later in Section 4.3.1, we show its connection to facial reduction (FR) through a simplification of the algorithm.

Let $P = \{\mathbf{x} \in \mathbb{R}^n \mid A\mathbf{x} \leq \mathbf{b}\}$ be a nonempty polyhedron. Fukuda's algorithm begins by solving the following linear program. In the initial iteration, $I = \varnothing$ and $J = \{1, \ldots, m\}$.

$$\begin{aligned} p_1^* := \max \quad & s \\ \text{s.t.} \quad & A_I\,\mathbf{x} = \mathbf{b}_I \\ & A_J\,\mathbf{x} + \mathbf{1}\,s \leq \mathbf{b}_J \\ & s \leq 1, \end{aligned} \tag{4.1}$$

If P is an empty set, then $p_1^* < 0$ and the algorithm terminates. Since P is nonempty, it is clear that $p_1^* \geq 0$. If $p_1^* > 0$, then there exists a point $\hat{\mathbf{x}} \in P$ such that $A_J\hat{\mathbf{x}} < \mathbf{b}_J$, which indicates that P is full-dimensional, and the algorithm terminates. If $p_1^* = 0$, then there is at least one implicit equality in $A_J\mathbf{x} \leq \mathbf{b}_J$.

To find the implicit equality when $p_1^* = 0$, the algorithm considers the dual problem

$$\begin{aligned} d_1^* := \min \quad & \mathbf{b}_I^T\mathbf{y}_I + \mathbf{b}_J^T\mathbf{y}_J + t \\ \text{s.t.} \quad & A_I^T\mathbf{y}_I + A_J^T\mathbf{y}_J = \mathbf{0} \\ & \mathbf{1}^T\mathbf{y}_J + t = 1 \\ & \mathbf{y}_J \geq \mathbf{0}, \quad t \geq 0. \end{aligned} \tag{4.2}$$

Let $(\mathbf{y}^*, t^*)$ be an optimal solution of (4.2) and $K_1 := \{j \in J \mid y_j^* > 0\}$. It can be shown that $K_1 \neq \emptyset$. (Indeed, if $\mathbf{y}_J^* = \mathbf{0}$, then the constraints in (4.2) indicate $t^* = 1$ and $A_I^T\mathbf{y}_I^* = \mathbf{0}$. Furthermore, if $\mathbf{y}_J^* = \mathbf{0}$, then strong duality implies $\mathbf{b}_I^T\mathbf{y}_I^* + t^* = d_1^* = p_1^* = 0$. Consequently, the assumption $P \neq \emptyset$ is violated since $0 = \mathbf{x}^T A_I^T\mathbf{y}_I^* = \mathbf{b}_I^T\mathbf{y}_I^* = -1$ for any $\mathbf{x} \in P$.) Now, note that when $p_1^* = 0$, $(\mathbf{x}, 0)$ is optimal to (4.1) for any $\mathbf{x} \in P$. Hence, for any $j \in K_1$ and $\mathbf{x} \in P$, complementary slackness $(b_j - a_j^T\mathbf{x})y_j^* = 0$ implies that $b_j - a_j^T\mathbf{x} = 0$. Therefore, $\mathbf{a}_j^T\mathbf{x} \leq b_j$ is an implicit equality for all $j \in K_1$. The algorithm then sets $I \leftarrow I \cup K_1$ and $J \leftarrow J \setminus K_1$. Next, note that for any $j \in J$, if $\begin{bmatrix} \mathbf{a}_j^T & b_j \end{bmatrix}$ is in the row space of $\begin{bmatrix} A_I & \mathbf{b}_I \end{bmatrix}$, then it is clear that $\mathbf{a}_j^T\mathbf{x} \leq b_j$ is an implicit equality implied by $A_I\mathbf{x} = \mathbf{b}_I$. Now, let

$$K_2 := \left\{ j \in J \;\middle|\; \begin{bmatrix} \mathbf{a}_j^T & b_j \end{bmatrix} \text{ is in the row space of } \begin{bmatrix} A_I & \mathbf{b}_I \end{bmatrix} \right\}.$$

Fukuda's algorithm identifies K_2 using Gaussian elimination and updates $I \leftarrow I \cup K_2$ and $J \leftarrow J \setminus K_2$. Fukuda's algorithm repeats the above processes until all implicit equalities are identified. Upon termination, I contains the indices of all of the implicit equalities, so $\text{aff}(P) = \{\mathbf{x} \in \mathbb{R}^n \mid A_I\mathbf{x} = \mathbf{b}_I\}$.

For convenience, we also provide Fukuda's algorithm in Algorithm 1. Due to numerical issues, we note that it can be difficult to determine whether or not a small value is zero. For this reason, instead of checking whether p^* or y_j^* is strictly positive in Algorithm 1, we use a small positive tolerance ϵ. Unfortunately, this is not guaranteed to resolve all numerical issues as it is also hard to determine a suitable tolerance.

Algorithm 1 Fukuda's Algorithm

Require: $P = \{\mathbf{x} \in \mathbb{R}^n \mid A\mathbf{x} \leq \mathbf{b}\}$

Initialize $I = \varnothing$ and $J = \{1, \ldots, m\}$

Solve (4.1)

while $|p^*| < \epsilon_s$ **do**

    Retrieve the optimal dual solution $\mathbf{y}^* = (\mathbf{y}_I^*, \mathbf{y}_J^*)$ to (4.2).

    Set $K_1 = \{j \in J \mid y_j^* > \epsilon\}$

    $I \leftarrow I \cup K_1$

    $J \leftarrow J \setminus K_1$

    Set $K_2 = \left\{j \in J \mid \begin{bmatrix}\mathbf{a}_j^T & b_j\end{bmatrix} \text{ is in the row space of } \begin{bmatrix}A_I & \mathbf{b}_I\end{bmatrix}\right\}$

    $I \leftarrow I \cup K_2$

    $J \leftarrow J \setminus K_2$

    Solve (4.1)

end while

4.2.2 Splitting method

Let $\|\cdot\|$ denote the zero norm. We now review a method for finding an element in a polyhedron with the maximum number of non-zeros from [45]. To lay the groundwork, we first consider

$$\begin{aligned} \max \quad & \|\mathbf{x}\| \\ \text{s.t.} \quad & \mathbf{x} \in C \end{aligned} \tag{4.3}$$

where C is a convex cone. We say C is *nonnegative* iff $\mathbf{x} \geq \mathbf{0}$ for all $\mathbf{x} \in C$.

The splitting method in [44] is founded on the optimization problem given in Lemma 2 that obtains a maximal element of a nonnegative convex cone. To simplify the explanation for how problem (4.5) in Lemma 2 can be used to get an optimal solution to problem (4.3), we will use the following lemma.

Lemma 1. *Consider the problem* (4.3)*. Assume* C *is non-negative. Then an optimal solution for* (4.3) *can be obtained by solving*

$$\begin{aligned} \max \quad & \mathbf{1}^T \min\{\mathbf{x}, \mathbf{1}\} \\ \text{s.t.} \quad & \mathbf{x} \in C \end{aligned} \tag{4.4}$$

Proof. Let $\mathbf{x}^* \geq 0$ be optimal for (4.4). As $\lambda \mathbf{x}^*$ is optimal for any $\lambda \geq 1$, we have $x_i^* > 0$ implies that $x_i^* \geq 1$. This yields $\|\mathbf{x}^*\| = \mathbf{1}^T \min\{\mathbf{x}^*, \mathbf{1}\}$. We claim that $\mathbf{x}^*$ is optimal for (4.3). If not, there exists a feasible $\mathbf{x}$ for (4.3) such that $\|\mathbf{x}\| > \|\mathbf{x}^*\|$. Then, $E := \{i \mid x_i > 0, x_i^* = 0\}$ is not empty. As C is a convex cone, $\mathbf{x}^* + \mathbf{x}$ is also feasible for (4.4). Moreover, we have $\mathbf{1}^T \min\{\mathbf{x}^* + \mathbf{x}, \mathbf{1}\} > \mathbf{1}^T \min\{\mathbf{x}^*, \mathbf{1}\}$ as $E \neq \emptyset$, which is a contradiction to the optimality of $\mathbf{x}^*$. □

Now that we have shown that problem (4.4) yields a maximal element to the non-negative convex cone C, we provide the following lemma which is the basis for the splitting method.

Lemma 2. *The problem* (4.4) *is equivalent to following problem*

$$\begin{aligned} \max \quad & \mathbf{1}^T \mathbf{u} \\ \text{s.t.} \quad & \mathbf{u} + \mathbf{v} \in C \\ & \mathbf{u} \leq \mathbf{1} \\ & \mathbf{u}, \mathbf{v} \geq 0 \end{aligned} \tag{4.5}$$

Proof. For any feasible $\mathbf{x}$ in (4.4), we define $\mathbf{u}$ and $\mathbf{v}$ as follows. Let $\mathbf{u} = \min\{\mathbf{x}, \mathbf{1}\}$ and $\mathbf{v} = \max\{\mathbf{x}, \mathbf{1}\} - \mathbf{1}$. Then, $\mathbf{u} + \mathbf{v} = \mathbf{x} \in C$, $\mathbf{0} \leq \mathbf{u} \leq \mathbf{1}$ and $\mathbf{v} \geq \mathbf{0}$. This shows $(\mathbf{u}, \mathbf{v})$ is a feasible solution for (4.5) with the same objective value as (4.4).

For any optimal $(\mathbf{u}, \mathbf{v})$ for (4.5), we have $\mathbf{x} = \mathbf{u} + \mathbf{v}$ is feasible for (4.4). Moreover, one can easily verify that $\mathbf{u} = \min\{\mathbf{x}, \mathbf{1}\}$ and thus they have the same objective values. □

Let (u_i^*, v_i^*) be an optimal solution to (4.5), and let $I^* := \{i \mid u_i^* + v_i^* > 0\}$. Note that if $i \in I^*$, then $u_i^* = 1$. Otherwise, we could scale $(\mathbf{u}^*, \mathbf{v}^*)$ by $\lambda > 0$ such that $\lambda(u_i^* + v_i^*) \geq 1$ for all $i \in I^*$. Then, we could define a feasible solution $(\hat{\mathbf{u}}, \hat{\mathbf{v}})$ such that

$$\hat{u}_i = \begin{cases} 1, & \text{if } i \in I^* \\ 0, & \text{otherwise} \end{cases} \qquad \text{and} \qquad \hat{v}_i = \begin{cases} \lambda(u_i^* + v_i^*) - 1, & \text{if } i \in I^* \\ 0, & \text{otherwise.} \end{cases} \tag{4.6}$$

Thus, if there exists $i \in I^*$ such that $u_i^* < 1$, then $1^T \hat{\mathbf{u}} > 1^T \mathbf{u}^*$, which would contradict the optimality of $(\mathbf{u}^*, \mathbf{v}^*)$.

Note that C can also be given as the projection of a higher dimension set. For example, $C = \{\mathbf{x} \in \mathbb{R}^n \mid A\mathbf{x} + B\mathbf{y} \leq \mathbf{0} \text{ for some } \mathbf{y}\}$. This includes a special case when we are interested in maximizing the non-zeros in a subset of entries. Using this idea and homogenization, Mehdiloozad et al. developed a method for finding the relative interior of a polyhedron by maximizing the non-zeros in the slack variables.

Let $P = \{\mathbf{x} \mid A\mathbf{x} \leq \mathbf{b}\}$ be a polyhedron. By introducing a non-negative auxiliary variable associated with the constant terms, P can be converted to a cone in a one dimensional higher space. Thus, the characteristic cone of P is defined as

$$P_+ := \{(x_0, \mathbf{x}) \in \mathbb{R} \times \mathbb{R}^n \mid \mathbf{b}x_0 - A\mathbf{x} \geq \mathbf{0},\ x_0 > 0\}. \tag{4.7}$$

Thus, $\mathbf{x} \in P$ if and only if $(1, \mathbf{x}) \in P_+$. Similarly, $(x_0, \mathbf{x}) \in P_+$ if and only if $\frac{1}{x_0}\mathbf{x} \in P$. After adding in slack variables, we have

$$\hat{P}_+ := \{(x_0, \mathbf{x}, \mathbf{s}) \in \mathbb{R} \times \mathbb{R}^n \times \mathbb{R}^m \mid \mathbf{b}x_0 - A\mathbf{x} - \mathbf{s} = \mathbf{0},\ x_0 > 0,\ \mathbf{s} \geq \mathbf{0}\}. \tag{4.8}$$

Consequently, $\mathbf{x} \in P$ if and only if $(1, \mathbf{x}, \mathbf{b} - A\mathbf{x}) \in \hat{P}_+$. The projection of the polyhedral cone $\hat{P}_+$ to the $\mathbf{s}$-space is

$$C_{\hat{P}_+} := \{\mathbf{s} \in \mathbb{R}^n \mid (x_0, \mathbf{x}, \mathbf{s}) \in \hat{P}_+\}. \tag{4.9}$$

Therefore, applying Lemma 2 to the cone $C_{\hat{P}_+}$ splits $\mathbf{s}$ and yields:

$$\begin{aligned} \max \quad & \mathbf{1}^T\mathbf{u} \\ \text{s.t.} \quad & \mathbf{b}x_0 - A\mathbf{x} - \mathbf{u} - \mathbf{v} = \mathbf{0} \\ & \mathbf{u} \leq \mathbf{1} \\ & \mathbf{u}, \mathbf{v} \geq \mathbf{0} \\ & x_0 > 0 \end{aligned} \tag{4.10}$$

However, since strict inequalities are not preferred, the splitting method instead uses the LP in the following lemma.

Lemma 3. *Maximization problem* (4.10) *can equivalently be formulated as*

$$\begin{aligned} \max \quad & \mathbf{1}^T\mathbf{u} \\ \text{s.t.} \quad & \mathbf{b}x_0 - A\mathbf{x} - \mathbf{u} - \mathbf{v} = \mathbf{0} \\ & \mathbf{u} \leq \mathbf{1} \\ & \mathbf{u}, \mathbf{v} \geq \mathbf{0} \\ & x_0 \geq 1 \end{aligned} \tag{4.11}$$

Proof. Note that the feasible region of (4.11) is a subset of the feasible region of (4.10). Thus, the optimal value of (4.11) is less than or equal to the optimal value of (4.10). Now, let $(x_0^*, \mathbf{x}^*, \mathbf{u}^*, \mathbf{v}^*)$ be an optimal solution to (4.10), and suppose $x_0^* \in (0,1)$. Define $\hat{\mathbf{x}} = \frac{1}{x_0^*}\mathbf{x}^*$, $\hat{\mathbf{u}} = \mathbf{u}^*$, and $\hat{\mathbf{v}}^* = \frac{1}{x_0^*}\mathbf{v}^* + \left(\frac{1}{x_0^*} - 1\right)\mathbf{u}^*$. Then, $(1, \hat{\mathbf{x}}, \hat{\mathbf{u}}, \hat{\mathbf{v}})$ is a feasible solution to (4.11) with an objective value of $\mathbf{1}^T\hat{\mathbf{u}} = \mathbf{1}^T\mathbf{u}^*$. Therefore, $(1, \hat{\mathbf{x}}, \hat{\mathbf{u}}, \hat{\mathbf{v}})$ is optimal to (4.11). □

Finally, we will show that $\mathbf{a}_i^T\mathbf{x} \leq b_i$ is an implicit equality iff $u_i^* = 0$. Suppose $\mathbf{a}_i^T\mathbf{x} \leq b_i$ is an implicit equality. Then, for all $\mathbf{x} \in P$, $\mathbf{a}_i^T\mathbf{x} = b_i$. Similarly, for all $(x_0, \mathbf{x}) \in P_+$, $b_i x_0 - \mathbf{a}_i^T\mathbf{x} = 0$. Thus, for all $(x_0, \mathbf{x}, \mathbf{s}) \in \hat{P}_+$, $s_i = 0$. Since $s_i = u_i + v_i$, this implies that $u_i = v_i = 0$ for all feasible $(x_0, \mathbf{x}, \mathbf{u}, \mathbf{v})$. Therefore, $u_i^* = 0$. Now, let $(x_0^*, \mathbf{x}^*, \mathbf{u}^*, \mathbf{v}^*)$ be an optimal solution to (4.11), and suppose that $u_i^* = 0$. Then, $v_i^* = 0$. (Otherwise, $(x_0^*, \mathbf{x}^*, \mathbf{u}^* + \min\{v_i^*, 1\}\mathbf{e}_i, \mathbf{v}^* - \min\{v_i^*, 1\}\mathbf{e}_i)$ is a feasible solution to (4.11) with a greater objective value.) Thus, $b_i x_0^* - \mathbf{a}_i^T \mathbf{x}^* = 0$. Suppose by contradiction that there exists $\mathbf{x} \in P$ s.t. $\mathbf{a}_i^T \mathbf{x} < b_i$. Then, there exists a feasible $(x_0, \mathbf{x}, \alpha \mathbf{e}_i, \mathbf{v})$ s.t. $\alpha > 0$. Hence, $(x_0^*, \mathbf{x}^*, \mathbf{u}^*, \mathbf{v}^*) + (x_0, \mathbf{x}, \alpha \mathbf{e}_i, \mathbf{v})$ is also feasible to (4.11) and $\mathbf{1}^T(\mathbf{u}^* + \alpha \mathbf{e}_i) = \mathbf{1}^T \mathbf{u}^* + \alpha$, which contradicts the optimality of $(x_0^*, \mathbf{x}^*, \mathbf{u}^*, \mathbf{v}^*)$. Therefore, $\mathbf{a}_i^T \mathbf{x} \leq b_i$ is an implicit equality.

4.2.3 Freund's LPs

In [23], Freund et al. give a few LPs that can be used to determine the implicit inequalities in a linear system $A\mathbf{x} \leq \mathbf{b}$. Freund et al. note that the first two methods, denoted by (P1) and (P2), are very closely related; however, (P1) necessitates a sufficiently small parameter ε whose value can be difficult to determine. For this reason, we will not focus on (P1), but we will look at at the other two methods.

4.2.3.1 P2 method

Thus, the first of Freund et al.'s methods that we look at is (P2):

$$\begin{aligned} \max \quad & \mathbf{1}^T \mathbf{s} \\ \text{s.t.} \quad & A\mathbf{x} + \mathbf{s} - \mathbf{b}\alpha \leq \mathbf{0} \\ & \mathbf{0} \leq \mathbf{s} \leq \mathbf{1} \\ & \alpha \geq 1 \end{aligned} \tag{P2}$$

To determine which of the inequalities are implicit equalities using the optimal solution of (P2), Freund et al. give the following proposition without proof; for the sake of completeness, we include a proof.

Proposition 1. *[23] If the system* $A\mathbf{x} \leq \mathbf{b}$ *is feasible, then* (P2) *is feasible and finite, and for any optimal solution* $(\mathbf{x}^*, \mathbf{s}^*, \alpha^*)$ *to* (P2)*, the set of indices of implicit equalities is given by* $\{i \mid s_i^* = 0\}$*. Furthermore,* $\frac{1}{\alpha^*}\mathbf{x}^*$ *is an element of the relative interior of* $\{\mathbf{x} \mid A\mathbf{x} \leq \mathbf{b}\}$*. If the system* $A\mathbf{x} \leq \mathbf{b}$ *is infeasible, then* (P2) *is infeasible.*

Proof.

- Let A be an $m \times n$ matrix. Suppose $A\mathbf{x} \leq \mathbf{b}$ is feasible, and let $\hat{\mathbf{x}}$ be a solution. Then, $(\mathbf{x}, \mathbf{s}, \alpha) = (\hat{\mathbf{x}}, \mathbf{0}, 1)$ is a solution to (P2). Moreover, since $\mathbf{0} \leq \mathbf{s} \leq \mathbf{1}$ for any feasible solution to (P2), the optimal value of (P2) lies in $[0, m]$. Thus, if the system $A\mathbf{x} \leq \mathbf{b}$ is feasible, then (P2) is feasible and finite.

- Let S denote the set of indices of implicit equalities in $A\mathbf{x} \leq \mathbf{b}$. Then, for any feasible solution $(\mathbf{x}, \mathbf{s}, \alpha)$ to (P2), $s_i = 0$ for all $i \in S$. If not, then $(\frac{1}{\alpha}\mathbf{x}, \frac{1}{\alpha}\mathbf{s}, 1)$ is a feasible solution to (P2) such that $\mathbf{a}_i^T(\frac{1}{\alpha}\mathbf{x}) < b_i$, which contradicts $i \in S$. Thus, $S \subseteq \{i \mid s_i^* = 0\}$.

  Now, let $(\mathbf{x}^*, \mathbf{s}^*, \alpha^*)$ be an optimal solution of (P2), and suppose there exists $i \in \{i \mid s_i^* = 0\} \setminus S$. Then, there exists $\hat{\mathbf{x}}$ such that $A\hat{\mathbf{x}} \leq \mathbf{b}$ and $\mathbf{a}_i^T\hat{\mathbf{x}} < b_i$. Consequently, $(\hat{\mathbf{x}}, \beta\mathbf{e}_i, 1)$ is a feasible solution to (P2), where $0 < \beta \leq \min\{b_i - \mathbf{a}_i^T\hat{\mathbf{x}},\ 1\}$. Then, it can be seen that $(\mathbf{x}^* + \hat{\mathbf{x}}, \mathbf{s}^* + \beta\mathbf{e}_i, 1 + \alpha^*)$ is a feasible solution to (P2) with an objective value of $\mathbf{1}^T\mathbf{s}^* + \beta$, which contradicts the optimality of $(\mathbf{x}^*, \mathbf{s}^*, \alpha^*)$. Thus, $\{i \mid s_i^* = 0\} \subseteq S$. Therefore, $S = \{i \mid s_i^* = 0\}$; i.e., the set of indices of implicit equalities is given by $\{i \mid s_i^* = 0\}$.

- Let $(\mathbf{x}^*, \mathbf{s}^*, \alpha^*)$ be an optimal solution to (P2). We will show that $\frac{1}{\alpha^*}\mathbf{x}^*$ is an element of the relative interior of $\{\mathbf{x} \mid A\mathbf{x} \leq \mathbf{b}\}$. To begin, note that for all $i \in \{i \mid \mathbf{s}_i^* = 0\}$, $\mathbf{a}_i^T(\frac{1}{\alpha^*}\mathbf{x}^*) = b_i$. (Otherwise, there exists $\beta > 0$ such that $(\mathbf{x}^*, \mathbf{s}^* + \beta\mathbf{e}_i, \alpha^*)$ is a feasible solution to (P2) whose objective value is $\mathbf{1}^T\mathbf{s}^* + \beta > \mathbf{1}^T\mathbf{s}^*$.) In contrast, for all $i \notin \{i \mid s_i^* = 0\}$, $\mathbf{a}_i^T\mathbf{x}^* < \alpha^* b_i$, so $\mathbf{a}_i^T(\frac{1}{\alpha^*}\mathbf{x}^*) < b_i$. Therefore, since we have already shown that $s_i^* = 0$ iff $\mathbf{a}_i^T\mathbf{x} \leq b_i$ is an implicit equality, this guarantees $\frac{1}{\alpha^*}\mathbf{x}^*$ is an element of the relative interior of $\{\mathbf{x} \mid A\mathbf{x} \leq \mathbf{b}\}$.

- Suppose $A\mathbf{x} \leq \mathbf{b}$ is infeasible. Let $(\mathbf{x}, \mathbf{s}, \alpha)$ be a solution to (P2). Then, $A\mathbf{x} + \mathbf{s} - \mathbf{b}\alpha \leq \mathbf{0}$. Since $\mathbf{s} \geq \mathbf{0}$, this implies $A\mathbf{x} - \mathbf{b}\alpha \leq \mathbf{0}$. Furthermore, since $\alpha \geq 1$, this implies that $A(\frac{1}{\alpha}\mathbf{x}) - \mathbf{b} \leq \mathbf{0}$. Hence, $\hat{\mathbf{x}} = \frac{1}{\alpha}\mathbf{x}$ is a solution to $A\mathbf{x} \leq \mathbf{b}$, which is a contradiction. Therefore, if the system $A\mathbf{x} \leq \mathbf{b}$ is infeasible, then (P2) is infeasible.

□

Note that in the more general case where the system is given by $A_I\mathbf{x} = \mathbf{b}_I, A_J\mathbf{x} \leq \mathbf{b}_J$, (P2) is then given by:

$$\begin{aligned} \max \quad & \mathbf{1}^T\mathbf{s} \\ \text{s.t.} \quad & A_I\mathbf{x} \qquad - \mathbf{b}_I\alpha = \mathbf{0} \\ & A_J\mathbf{x} + \mathbf{s} - \mathbf{b}_J\alpha \leq \mathbf{0} \\ & \mathbf{0} \leq \mathbf{s} \leq \mathbf{1} \\ & \alpha \geq 1 \end{aligned}$$

and Proposition 1 still holds.

4.2.3.2 P4 method

The final method given in [23] is based on finding a strictly complemntary pair of optimal solutions to the following primal and dual LPs:

$$\begin{aligned} \max \quad & s \\ \text{s.t.} \quad & A\mathbf{x} + \mathbf{1}s \leq \mathbf{b} \end{aligned} \qquad \text{(P3)}$$

$$\begin{aligned} \min \quad & \mathbf{b}^T\mathbf{y} \\ \text{s.t.} \quad & A^T\mathbf{y} = \mathbf{0} \\ & \mathbf{1}^T\mathbf{y} = 1 \\ & \mathbf{y} \geq \mathbf{0} \end{aligned} \qquad \text{(D3)}$$

Thus, the final LP in [23] is given by:

$$
\begin{aligned}
\max \quad & \theta \\
\text{s.t.} \quad & A\mathbf{x} + \mathbf{1}s \leq \mathbf{b} \\
& A^T\mathbf{y} = \mathbf{0} \\
& \mathbf{1}^T\mathbf{y} = 1 \\
& -\mathbf{b}^T\mathbf{y} + s = 0 \\
& \mathbf{y} - A\mathbf{x} - \mathbf{1}s - \mathbf{1}\theta \geq -\mathbf{b} \\
& \mathbf{y} \geq \mathbf{0}
\end{aligned}
\tag{P4}
$$

where the first constraint ensures primal feasibility, the second, third, and sixth constraint ensure dual feasibility, the fourth constraint ensures strong duality, and the fifth constraint ensures strict complementarity of $\mathbf{y}$ and $(\mathbf{b} - A\mathbf{x} - \mathbf{1}s)$.

Proposition 2. *[23] If* (P4) *is infeasible, then $A\mathbf{x} \leq \mathbf{b}$ has no implicit equalities. If* (P4) *is feasible, then it is finite, and for any optimal solution $(\mathbf{x}^*, \mathbf{y}^*, s^*, \theta^*)$, we have:*

(i) If $s^* = 0$, the set of indices of implicit equalities is given by $\{i \mid y_i^* > 0\}$, and $\mathbf{x}^*$ lies in the relative interior of $\{\mathbf{x} \mid A\mathbf{x} \leq \mathbf{b}\}$.

(ii) If $s^* > 0$, there are no implicit equalities.

(iii) If $s^* < 0$, the system $A\mathbf{x} \leq \mathbf{b}$ has no solutions.

Proof. Note that (P3) is always feasible as $\mathbf{x} = 0$ and $s = \min_i b_i$ is a feasible solution. Also note that that if (P3) has a finite optimal value, then (P4) is feasible by strong duality. On the other hand, if the optimal value of (P3) is unbounded, then (D3) and consequently (P4) will be in infeasible. Thus, (P4) is infeasible if and only if (P3) is unbounded. Furthermore, if the optimal value of (P3) is unbounded, then $\mathbf{b} - A\mathbf{x} > \mathbf{0}$ for some $\mathbf{x}$. Therefore, if $(P4)$ is infeasible, then $A\mathbf{x} \leq \mathbf{b}$ has no implicit equalities.

We will now show that if (P4) is feasible, then (P4) has a finite optimal value. Let $(\mathbf{x}, \mathbf{y}, s, \theta)$ be a feasible solution to (P4). Since $\mathbf{1}^T\mathbf{y} = 1$, there exists $i \in \{1, \ldots, m\}$ such that $0 < y_i \leq 1$. Hence, by the complementary slackness, we have $b_i - \mathbf{a}_i^T\mathbf{x} - s = 0$. Thus, by the complementarity slackness constraint, we have

$$\theta \leq (b_i - \mathbf{a}_i^T\mathbf{x} - s) + y_i = y_i \leq 1.$$

Therefore, if (P4) is feasible, then (P4) has a finite optimal value.

Finally, let $(\mathbf{x}^*, \mathbf{y}^*, s^*, \theta^*)$ be an optimal solution to (P4), and consider the following cases:

(i) Suppose $s^* = 0$. Since $(\mathbf{x}^*, s^*)$ is an optimal solution to (P3), this implies that $A\mathbf{x} \leq \mathbf{b}$ has at least one implicit inequality. (Otherwise, there exists feasible solutions $\mathbf{x}_1, \ldots, \mathbf{x}_k$ to $A\mathbf{x} \leq \mathbf{b}$ such that for all $i \in \{1, \ldots, m\}$, there exists $j \in \{1, \ldots, k\}$ such that $\mathbf{a}_i^T\mathbf{x}_j < b_i$. Hence, by taking a convex combination of these solutions, we obtain a feasible solution $\hat{\mathbf{x}} = \frac{1}{k}\sum_{j=1}^{k}\mathbf{x}_j$ such that $A\hat{\mathbf{x}} < \mathbf{b}$, which contradicts $s^* = 0$.) It now remains to show that the set of indices of implicit equalities is given by $\{i \mid y_i^* > 0\}$, and $\mathbf{x}^*$ lies in the relative interior of $\{\mathbf{x} \mid A\mathbf{x} \leq \mathbf{b}\}$.

Note that $\mathbf{y}^*$ is an optimal solution to (D3). Also, note that since $s^* = 0$, $(\mathbf{x}, s^*)$ is an optimal solution to $(P3)$ for any $\mathbf{x}$ such that $A\mathbf{x} \leq \mathbf{b}$. Thus, by complementary slackness, $(b_i - \mathbf{a}_i^T\mathbf{x})y_i^* = 0$ for all $i = 1, \ldots, m$ and all $\mathbf{x}$ such that $A\mathbf{x} \leq \mathbf{b}$. Thus, if $y_i^* > 0$, then $b_i - \mathbf{a}_i^T\mathbf{x} = 0$ for all x such that $A\mathbf{x} \leq \mathbf{b}$. Hence, $\mathbf{a}_i^T\mathbf{x} \leq b_i$ is an implicit equality. Similarly, by strict complementary slackness, if $y_i^* = 0$, then $b_i - \mathbf{a}_i^T\mathbf{x}^* > 0$. Therefore, the set of indices of implicit equalities is given by $\{i \mid y_i^* > 0\}$, and $\mathbf{x}^*$ lies in the relative interior of $\{\mathbf{x} \mid A\mathbf{x} \leq \mathbf{b}\}$.

(ii) Suppose $s^* > 0$. Then, $A\mathbf{x}^* < A\mathbf{x}^* + \mathbf{1}s^* \leq \mathbf{b}$, so $A\mathbf{x}^* < \mathbf{b}$. Thus, $A\mathbf{x} \leq \mathbf{b}$ has no implicit equalities.

(iii) Suppose $s^* < 0$. Also, suppose that $\hat{\mathbf{x}}$ is a solution to $A\mathbf{x} \leq \mathbf{b}$. Then, $(\hat{\mathbf{x}}, 0)$ is a solution (P3), which contradicts the fact that $s^* < 0$ is the optimal value of (P3). Therefore, if $s^* < 0$, then $A\mathbf{x} \leq \mathbf{b}$ has no solutions.

□

Note that since s is free, (P3) is always feasible. Thus, (P4) is only infeasible if (P3) is unbounded and (D3) is infeasible. Hence, if (P4) is infeasible, then $A\mathbf{x} \leq \mathbf{b}$ has no implicit equalities. Now, note that in the more general case where the system is given by $A_I\mathbf{x} = \mathbf{b}_I, A_J\mathbf{x} \leq \mathbf{b}_J$, then $A_I\mathbf{x} = \mathbf{b}_I$ is added as a primal feasibility constraint to (P3) and (P4). Similarly, the dual feasibility constraint $A^T\mathbf{y}$ in (D3) and (P4) is updated to $A_J^T\mathbf{y}_J + A_I^T\mathbf{y}_I = \mathbf{0}$. Moreover, the strong duality constraint in (P4) is updated to $-\mathbf{b}_J^T\mathbf{y}_J - \mathbf{b}_I^T\mathbf{y}_I + s = 0$. Thus, for a system given by $A_I\mathbf{x} = \mathbf{b}_I, A_J\mathbf{x} \leq \mathbf{b}_J$, we would solve the following LP:

$$\begin{aligned} \max \quad & \theta \\ \text{s.t.} \quad & A_J\mathbf{x} + \mathbf{1}s \leq \mathbf{b}_J \\ & A_I\mathbf{x} = \mathbf{b}_I \\ & A_J^T\mathbf{y}_J + A_I^T\mathbf{y}_I = \mathbf{0} \\ & \mathbf{1}^T\mathbf{y}_J = 1 \\ & -\mathbf{b}_J^T\mathbf{y}_J - \mathbf{b}_I^T\mathbf{y}_I + s = 0 \\ & \mathbf{y}_J - A_J\mathbf{x} - \mathbf{1}s - \mathbf{1}\theta \geq -\mathbf{b}_J \\ & \mathbf{y}_J \geq \mathbf{0} \end{aligned} \tag{$\overline{\text{P4}}$}$$

Now, note that with the addition of $A_I\mathbf{x} = \mathbf{b}_I$, (P3) is no longer guaranteed to be feasible, so if $(\overline{\text{P4}})$ is infeasible, it could mean that $A_J\mathbf{x} \leq \mathbf{b}_J$, $A_I\mathbf{x} = \mathbf{b}_I$ has no implicit equalities or it could mean that the system is infeasible. Thus, we give the following proposition for the case where the system is given by $A_I\mathbf{x} = \mathbf{b}_I, A_J\mathbf{x} \leq \mathbf{b}_J$.

Proposition 3. *If* $(\overline{\text{P4}})$ *is infeasible, then either the system* $A_J\mathbf{x} \leq \mathbf{b}_J$, $A_I\mathbf{x} = \mathbf{b}_I$ *is infeasible or the system is feasible but has no implicit equalities. If* $(\overline{\text{P4}})$ *is feasible, then it* *is finite, and for any optimal solution* $(\mathbf{x}^*, \mathbf{y}^*, s^*, \theta^*)$, *we have:*

(i) If $s^* = 0$, *the set of indices of implicit equalities is given by* $\{i \mid y_i^* > 0\}$, *and* $\mathbf{x}^*$ *lies in the relative interior of* $\{\mathbf{x} \mid A_J\mathbf{x} \leq \mathbf{b}_J,\ A_I\mathbf{x} = \mathbf{b}_I\}$.

(ii) If $s^* > 0$, *there are no implicit equalities.*

(iii) If $s^* < 0$, *the system* $A_J\mathbf{x} \leq \mathbf{b}_J$, $A_I\mathbf{x} = \mathbf{b}_I$ *has no solutions.*

4.3 Modifications to existing methods

4.3.1 Fukuda's method simiplification and relation to facial reduction

Fukuda's algorithm can be simplified by observing that the constraint $s \leq 1$ is unessential in (4.1). Consider

$$\begin{aligned} p_2^* := \max \quad & s \\ \text{s.t.} \quad & A_I\,\mathbf{x} \qquad = \mathbf{b}_I \\ & A_J\,\mathbf{x} + \mathbf{1}\,s \leq \mathbf{b}_J. \end{aligned} \tag{4.12}$$

It is clear that $p_2^* = 0$ if and only if $p_1^* = 0$. If $p_1^* > 0$, then $\operatorname{aff}(P) = \{\mathbf{x} \in \mathbb{R}^n \mid A_I\mathbf{x} = \mathbf{b}_I\}$. If $p_2^* = 0$, we consider the dual problem of (4.12)

$$\begin{aligned} d_2^* := \min \quad & \mathbf{b}_I^T\mathbf{y}_I + \ \mathbf{b}_J^T y_J \\ \text{s.t.} \quad & A_I^T\mathbf{y}_I + A_J^T\mathbf{y}_J = \mathbf{0} \\ & \mathbf{1}^T\mathbf{y}_J = 1 \\ & \mathbf{y}_J \geq \mathbf{0}. \end{aligned} \tag{4.13}$$

In general, we look at the dual variable y_j to determine if the jth primal constraint $a_j^T\mathbf{x} \leq b_j$ is an implicit equality. Let $(\mathbf{x}^*, s^*)$ be an optimal solution to (4.12) and $\mathbf{y}^*$ be an optimal solution to (4.13). If $s^* = 0$, then $y_j^* > 0$ implies $b_j - \mathbf{a}_j^T\mathbf{x}^* = 0$ by complimentary slackness. Since every feasible $\mathbf{x}$ is optimal when $s^* = 0$, this implies that $a_j^T\mathbf{x} \leq b_j$ is an implicit equality. (When $s > 0$, $A_J\mathbf{x} < \mathbf{b}_J$, so there are no implicit equalities. Consequently, if $s^* > 0$, $y_j^* > 0$ does not indicate an implicit equality; it just means that $\mathbf{a}_j^T\mathbf{x}^* + s^* = b_j$.)

4.3.1.1 Relation to facial reduction

Finding an optimal solution $\mathbf{y}^*$ is indeed related to facial reduction (FR) of the nonnegative orthant. The FR procedure is a conceptual method that discovers the minimal face of a convex cone containing a feasible region [19]. (Recall that a set $\mathcal{F}$ is a *face* of the non-negative orthant $\mathbb{R}^n_+$ if there exists $J \subseteq \{1, \ldots, n\}$ such that $\mathcal{F} = \{\mathbf{x} \in \mathbb{R}^n_+ \mid x_i = 0 \text{ for all } i \in J\}$.) In [19], it is shown that a feasible set $\{\mathbf{x} \in \mathbb{R}^n \mid A\mathbf{x} \leq \mathbf{b}\}$ is strictly feasible if and only if the auxiliary system

$$\mathbf{0} \neq \mathbf{y} \geq \mathbf{0},\ A^T\mathbf{y} = \mathbf{0},\ \mathbf{b}^T\mathbf{y} = 0 \tag{4.14}$$

is inconsistent. In fact, for $I = \emptyset$, $\bar{\mathbf{y}}$ is a solution to (4.14) if and only if $p_2^* = 0$ and $\bar{\mathbf{y}}/(\mathbf{1}^T\bar{\mathbf{y}})$ is an optimal solution of (4.13). (We ignore terms related to $\mathbf{y}_I$ and replace $\mathbf{y}_J$ with $\mathbf{y}$.) In general, we have the following equivalence.

Lemma 4. *Suppose that* $P = \{\mathbf{x} \in \mathbb{R}^n \mid A_I\mathbf{x} = \mathbf{b}_I,\ A_J\mathbf{x} \leq \mathbf{b}_J\}$ *is nonempty. Then,* $\bar{\mathbf{y}} = (\bar{\mathbf{y}}_I, \bar{\mathbf{y}}_J)$ *is a solution to*

$$\mathbf{0} \neq \mathbf{y}_J \geq \mathbf{0},\ A_I^T\mathbf{y}_I + A_J^T\mathbf{y}_J = \mathbf{0},\ \mathbf{b}_I^T\mathbf{y}_I + \mathbf{b}_J^T\mathbf{y}_J = 0 \tag{4.15}$$

if and only if $p_2^* = 0$ *and* $\bar{\mathbf{y}}/(\mathbf{1}^T\bar{\mathbf{y}}_J)$ *is an optimal solution of* (4.13).

Proof. The "if" direction is straightforward to check. For the "only if" direction, note that $p_2^* \geq 0$ due to $P \neq \emptyset$, and $\bar{\mathbf{y}}/(\mathbf{1}^T\bar{\mathbf{y}}_J)$ is feasible to (4.13) with objective value 0 when $\bar{\mathbf{y}}$ is feasible to (4.15). □

According to Lemma 4, instead of solving (4.1) and (4.2), one can solve the feasibility problem (4.15) in Fukuda's algorithm. If (4.15) is inconsistent, then $\operatorname{aff}(P) = \{\mathbf{x} \in$

$\mathbb{R}^n \mid A_I\mathbf{x} = \mathbf{b}_I\}$. Otherwise, implicit equalities can be identified based on the positivity of the components of a solution to (4.15) in the same way as in the original Fukuda's algorithm.

4.3.1.2 Fukuda's method simplification utilizing the dual

Due to numerical issues, one of the challenges of using Fukuda's algorithm in practice is determining whether or not s is positive. To avoid having to make that call ourselves, we can instead utilize Fukuda's dual, and let the solver decide whether or not the problem is feasible. (When solving the dual (4.13), we take the objective and add it to the constraints by requiring $\mathbf{b}_I^T\mathbf{y}_I + \mathbf{b}_J^T\mathbf{y}_J = 0$.) If this new problem is infeasible, then $s > 0$ in the primal, and there are no implicit equalities. If it is feasible, then $\mathbf{1}^T\mathbf{y} = 1$ yields positive entries in $\mathbf{y}$, which are hopefully not too small. If every y_i is still small, we can conclude at the very least that the largest y_i must be positive because if it is considered as zero in whichever criteria, then all the other entries must be zero, which would contradict the fact that they sum to 1.

Therefore, using this method, we set $K := J$ and solve:

$$\begin{aligned} d_2^* := \min \quad & 0 \\ \text{s.t.} \quad & A_I^T\mathbf{y}_I + A_J^T\mathbf{y}_J = \mathbf{0} \\ & \mathbf{b}_I^T\mathbf{y}_I + \mathbf{b}_J^T\mathbf{y}_J = 0 \\ & \mathbf{1}^T\mathbf{y}_K = 1 \\ & \mathbf{y}_K \geq \mathbf{0}. \end{aligned} \tag{4.16}$$

Then, we iteratively update $K = K \setminus \{k \in K \mid y_k^* > 0\} \subseteq J$ and resolve (4.16). We continue to update K and to resolve until (4.16) becomes infeasible, indicating there are no implicit equalities left to find.

4.3.1.3 What does putting objective as constraint mean?

Once again, consider the simplified LP that Fukuda's algorithm iteratively solves:

$$
\begin{aligned}
p_2^* := \max \quad & s \\
\text{s.t.} \quad & A_I\,\mathbf{x} \qquad\quad = \mathbf{b}_I \\
& A_J\,\mathbf{x} + \mathbf{1}\,s \le \mathbf{b}_J.
\end{aligned}
\tag{4.17}
$$

and its dual:

$$
\begin{aligned}
d_2^* := \min \quad & \mathbf{b}^T\mathbf{y} \\
\text{s.t.} \quad & -A^T\mathbf{y} = \mathbf{0} \\
& \mathbf{1}^T\mathbf{y}_J = 1 \\
& \mathbf{y}_J \ge \mathbf{0}.
\end{aligned}
\tag{4.18}
$$

where $A^T = [A_I^T \quad A_J^T]$, $\mathbf{b} = (\mathbf{b}_I, \mathbf{b}_J)$, and $\mathbf{y} = (\mathbf{y}_I, \mathbf{y}_J)$.

When solving the dual, we solve the following problem to avoid the discussion of whether or not the objective is zero.

$$
\begin{aligned}
\min \quad & 0 \\
\text{s.t.} \quad & -A^T\mathbf{y} = \mathbf{0} \\
& \mathbf{b}^T\mathbf{y} = 0 \\
& \mathbf{1}^T\mathbf{y}_J = 1 \\
& \mathbf{y}_J \ge \mathbf{0}.
\end{aligned}
\tag{4.19}
$$

Looking at its dual, we see that this corresponds to the homogenization of the primal problem:

$$
\begin{aligned}
\max \quad & s \\
\text{s.t.} \quad & A_I\,\mathbf{x} \qquad\quad = t\mathbf{b}_I \\
& A_J\,\mathbf{x} + \mathbf{1}\,s \le t\mathbf{b}_J.
\end{aligned}
\tag{4.20}
$$

Therefore, putting the objective as a constraint in the dual problem is the same as homogenizing the primal problem.

4.3.2 The splitting method applied to the dual approach

Consider the following primal-dual LPs:

$$\begin{aligned} \max \quad & 0 \\ \text{s.t.} \quad & A\mathbf{x} \leq \mathbf{b} \end{aligned} \qquad\qquad \begin{aligned} \min \quad & \mathbf{b}^T\mathbf{y} \\ \text{s.t.} \quad & A^T\mathbf{y} = \mathbf{0} \\ & \mathbf{y} \geq \mathbf{0} \end{aligned}$$

Note that if $A\mathbf{x} \leq \mathbf{b}$ is consistent, then any dual optimal solution $\mathbf{y}$ satisfies $\mathbf{b}^T\mathbf{y} = 0$ by strong duality. Thus, the dual problem can equivalently be written as a feasibility problem by adding $\mathbf{b}^T\mathbf{y} = 0$ as a constraint. Thus, we can convert the problem of finding $\text{aff}(P)$ to an equivalent problem of finding an optimal dual solution $\mathbf{y}^*$ whose support is maximal. The latter problem is an l_0-norm maximization problem over a non-negative polyhedral set,

$$\begin{aligned} \max \quad & \|\mathbf{y}\|_0 \\ \text{s.t.} \quad & \begin{bmatrix} A^T \\ \mathbf{b}^T \end{bmatrix} \mathbf{y} = \mathbf{0} \\ & \mathbf{y} \geq \mathbf{0} \end{aligned}$$

Note that unlike the feasible region defined by $A\mathbf{x} \leq \mathbf{b}$, the feasible region of the above problem is a nonnegative convex cone. Thus, it is very easy to directly apply Lemma 2 from Section 4.2.2 in order to get the following LP:

$$\begin{aligned} \max \quad & \mathbf{1}^T\mathbf{u} \\ \text{s.t.} \quad & \begin{bmatrix} A^T & A^T \\ \mathbf{b}^T & \mathbf{b}^T \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ \mathbf{v} \end{bmatrix} = \mathbf{0} \\ & \mathbf{u} \leq \mathbf{1} \\ & \mathbf{u}, \mathbf{v} \geq \mathbf{0} \end{aligned}$$

where $\mathbf{a}_i^T\mathbf{x} \leq b_i$ is an implicit eqaulity of P iff $u_i^* = 1$.

Now, recall that the values of the dual variables $\mathbf{u}$ and $\mathbf{v}$ are only meaningful if

$A\mathbf{x} \leq \mathbf{b}$ is consistent. The above maximization problem is always feasible as $(\mathbf{u}, \mathbf{v}) = (\mathbf{0}, \mathbf{0})$ is a feasible solution, and it always has a finite optimal value. Therefore, if one is unable to assume optimality, the feasibility problem $\min\{0 \mid A\mathbf{x} \leq \mathbf{b}\}$ should be solved before using the above maximization problem to check for implicit equalities.

4.3.3 Simplied P4 theory

In Section 4.2.3, we looked at two methods: (P2) and (P4); however, we observe that (P2) is equivalent to the primal splitting method (4.11) as the only difference is (4.11) includes slack variables. For this reason, we now focus on improving (P4).

Note that if we drop t from (P3), then we are able to get a simplified formulation for (P4). Thus, consider

$$\begin{array}{ll} \max & 0 \\ \text{s.t.} & A\mathbf{x} \leq \mathbf{b} \end{array} \qquad \text{(P3')} \qquad\qquad \begin{array}{ll} \min & \mathbf{b}^T\mathbf{y} \\ \text{s.t.} & A^T\mathbf{y} = \mathbf{0} \\ & \mathbf{y} \geq \mathbf{0} \end{array} \qquad \text{(D3')}$$

Then, the simplified formulation is given by

$$\begin{array}{rl} \max & \theta \\ \text{s.t.} & A\mathbf{x} \leq \mathbf{b} \\ & A^T\mathbf{y} = \mathbf{0} \\ & \mathbf{b}^T\mathbf{y} = 0 \\ & \mathbf{y} - A\mathbf{x} - \theta\mathbf{1} \geq -\mathbf{b} \\ & \mathbf{y} \geq \mathbf{0} \end{array} \qquad \text{(P4')}$$

Proposition 4. *If* (P4') *is infeasible, then* $A\mathbf{x} \leq \mathbf{b}$ *has no solution. If* (P4') *has a finite optimal value, then the set of indices of implicit equalities is given by* $\{i \mid y_i^* > 0\}$*. If* (P4') *is unbounded, then the set of indices of implicit equalities is given by* $\{i \mid \tilde{y}_i > 0\}$*,* *where* $(\mathbf{x}, \mathbf{y}, \theta)$ *is a feasible point and* $\tilde{\boldsymbol{r}} = (\tilde{\mathbf{x}}, \tilde{\mathbf{y}}, \tilde{\theta})$ *is an extreme ray such that such that* $(\mathbf{x}, \mathbf{y}, \theta) + \alpha\tilde{\boldsymbol{r}}$ *is feasible for all* $\alpha \geq 0$ *and* $\theta + \alpha\tilde{\theta} \to \infty$ *as* $\alpha \to \infty$*.*

While this simplified formulation is not significantly smaller than the original (P4) formulation, it has the advantage of not requiring us to judge whether or not a small s^* is zero. Also, note that in the more general case where the system is given by $A_I\mathbf{x} = \mathbf{b}_I, A_J\mathbf{x} \leq \mathbf{b}_J$, Proposition 4 still holds.

4.4 Numerical experiments

We implement each of the algorithms in Matlab 9.5 (R2018b) using Gurobi 9.0 to model and solve the corresponding LPs using Gurobi's concurrent method. We set a time limit of 15 minutes. Since this only limits Gurobi's runtime and the two Fukuda's algorithms are the only iterative methods, we also impose a time limit of one hour for the each of their functions. We run the instances on an Intel(R) Core(TM) i7-8550U CPU @ 1.80GHz with four cores and 16GB of memory.

For our numerical experiments, we use a set of benchmark MIP instances that we relax by removing the integrality constraints, and we also use some randomly generated instances. For the MIP instance, we use version 2 of the benchmark set from the Mixed-Integer Programming Library (MIPLIB) [25]. To generate instances, we use Algorithm 2 to generate polyhedrons. We run the algorithm with input combinations from $n \in \{100, 500, 1000\}$, $m \in \{100, 500, 1000, 5000\}$, $r \in \{0, 5, 25, 50, 250\}$, $p \in \{0.25, 0.5, 0.75\}$, and $m_q \in \{\min\{qm, qn\} \mid q = 0, 0.25, \text{ or } 0.5\}$. Thus, we generates nearly 400 feasible instances.

Algorithm 2 Generate random polyhedron

Require: n, m, m_q, p, r

Initialize $k = 0$
repeat
  $k \leftarrow k + 1$
  Generate an $(m - r) \times n$ matrix A with density p of uniformly distributed random numbers in the interval $(0, 1)$.
  Generate a vector $\mathbf{b}$ of length $(m - r)$ of uniformly distributed random numbers in the interval $(0, n)$.
  $A_I \leftarrow A(1 : m_q, :)$
  $\mathbf{b}_I \leftarrow \mathbf{b}(1 : m)$
  $A_J \leftarrow A(m_q + 1 : end, :)$
  $\mathbf{b}_J \leftarrow \mathbf{b}(m_q + 1 : end)$
  Solve $\min 0 \; s.t. \; A_J\mathbf{x} \leq \mathbf{b}_J, A_I\mathbf{x} = \mathbf{b}_I, \mathbf{x} \geq -n$.
until the above problem is feasible or $k \geq 15$
$k \leftarrow 0$
if $m_q = 0$ **then**
  repeat
    $k \leftarrow k + 1$
    Randomly generate a vector idx of length r of random integers in $\{1, \ldots, m - r\}$.
    Randomly generate an $r \times r$ matrix B with density 0.5 of uniformly distributed random numbers in the interval $(-10, 0)$.
    $B \leftarrow \text{floor}(B)$
    $\tilde{A}_J \leftarrow [A_J; BA_J(idx, :)]$ and $\tilde{\mathbf{b}}_J \leftarrow [\mathbf{b}_J; B\mathbf{b}_J(idx)]$
    Solve $\min 0 \; s.t. \; \tilde{A}_J\mathbf{x} \leq \tilde{\mathbf{b}}_J, A_I\mathbf{x} = \mathbf{b}_I, \mathbf{x} \geq -n$.
  until the above problem is feasible or $k \geq 15$.
else
  Randomly generate a vector idx of length r of random integers in $\{1, \ldots, m_q\}$.
  Randomly generate an $r \times r$ matrix B with density 0.5 of uniformly distributed random numbers in the interval $(-10, 0)$.
  $B \leftarrow \text{floor}(B)$
  $\tilde{A}_J \leftarrow [A_J; BA_I(idx, :)]$ and $\tilde{b}_J \leftarrow [\mathbf{b}_J; B\mathbf{b}_I(idx)]$
end if
$A_J \leftarrow \tilde{A}_J$
$\mathbf{b}_J \leftarrow \tilde{\mathbf{b}}_J$
if $\{\mathbf{x} \mid A_J\mathbf{x} \leq \mathbf{b}_J, A_I\mathbf{x} = \mathbf{b}_I, \mathbf{x} \geq -n\} \neq \varnothing$ **then**
  Save $A_J, A_I, \mathbf{b}_J, \mathbf{b}_I$
end if

The two tables below summarize the performance of each algorithm on the benchmark MIP instances (with integrality constraints removed) and on the randomly generated instances. Each table gives the number of instances for which the algorithm correctly counted the number of implicit equalities, the number of instances where an incorrect count was returned for the number of implicit equalities, the number of instances where a numerical issue prevented a count from being returned, and the number of instances that were not solved within the 15 minute time limit. We consider an instance to have numerical issues if Gurobi returned a "numeric" status due to unrecoverable numerical difficulties or if Gurobi returned an "unbounded" status when we know theoretically that the problem is bounded or if Gurobi returned an "infeasible" status when we have already determined the problem to be feasible or if Gurobi did not return a status (i.e., status was empty).

method	correct counts	incorrect counts	numerical issues	time limit
Fukuda original	197	0	0	43
Fukuda dual	191	0	0	49
Split primal	233	0	0	7
Split dual	227	0	0	13
P4 original	165	3	3	69
P4 simplified	180	3	1	56

Table 4.1: Benchmark Results. Summarizes the performance of each algorithm for the 240 benchmark instances. Of the 240 instances, there are five that were not solved by any of the algorithms within the 15 minute time limit.

method	correct counts	incorrect counts	numerical issues	time limit
Fukuda original	353	8	16	12
Fukuda dual	348	36	0	5
Split primal	389	0	0	0
Split dual	382	0	7	0
P4 original	355	11	4	19
P4 simplified	364	7	0	18

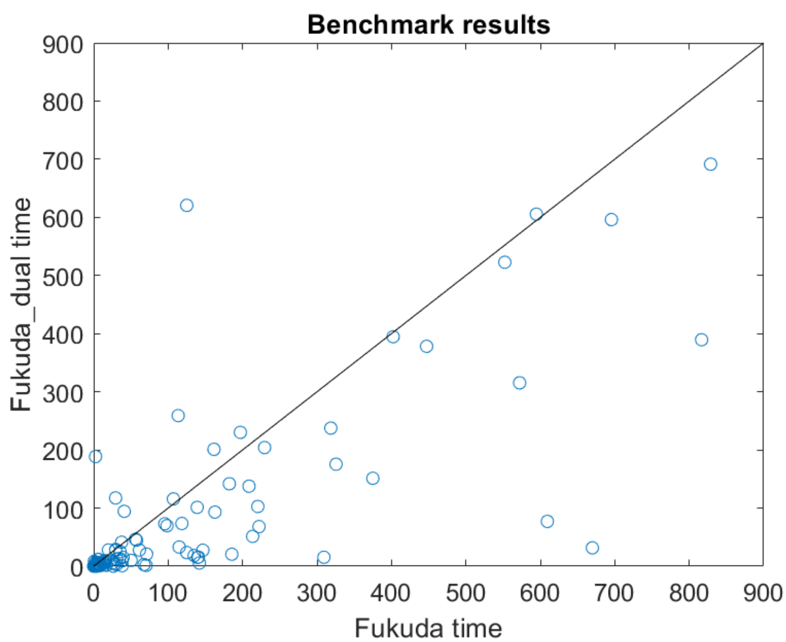
Table 4.2: Random Results. Summarizes the performance of each algorithm for the 389 randomly generated instances.

In the next few sections, we will compare each original method with the modified version that we propose. For the instances where both versions return the correct number of implicit inequalities, we give a graph comparing their gurobi runtimes.

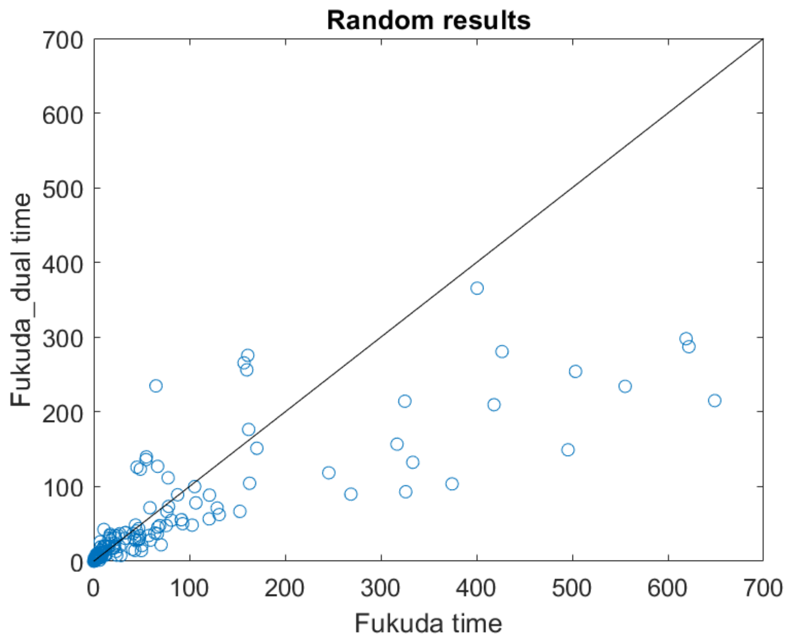
4.4.1 Fukuda

In our code for the original Fukuda's alogorithm, we used a tolerance of 10^{-6} to determine whether s is positive, negative, or zero. In both the original Fukuda alogirithm and our modified Fukuda's algorithm, we use a tolerance of 10^{-8} to determine whether or not a dual variable is positive.

Referring back to Table 4.1, we see that both versions reached the time limit for roughly 20% of the benchmark instances; however, our modified Fukuda's algorithm reached the time limit slightly more often than the original algorithm. There were 38 instances that neither version could solve within the time limit. Additionally, there were 11 instances that original Fukuda's algorithm solved within the time limit that modified Fukuda's algorithm did not, and there were five instances that modified Fukuda's algorithm solved within the time limit that original Fukuda's did not. For the remaining 186 instances that each algorithm was able to solve within the time limit, we give the following graph:



Referring back to Table 4.2, we see that original Fukuda's algorithm hit the time limit for about twice as many instances as our modified Fukuda's algorithm; however, our modified Fukuda is less reliable as it had more than four times as many instances where it returned an incorrect number of implicit inequalities. Other than incorrect counts, our modified Fukuda's algorithm did not suffer from any numerical issues whereas the original Fukuda's algorithm sometime ran into an issue where the initial iteration indicated the presence of implicit inequalities but the algorithm terminated when a later iteration indicated an infeasible status instead of a positive s as expected. Given these issues, there were 334 instances that were solved correctly within the time limit by both versions of Fukuda's algorithm.

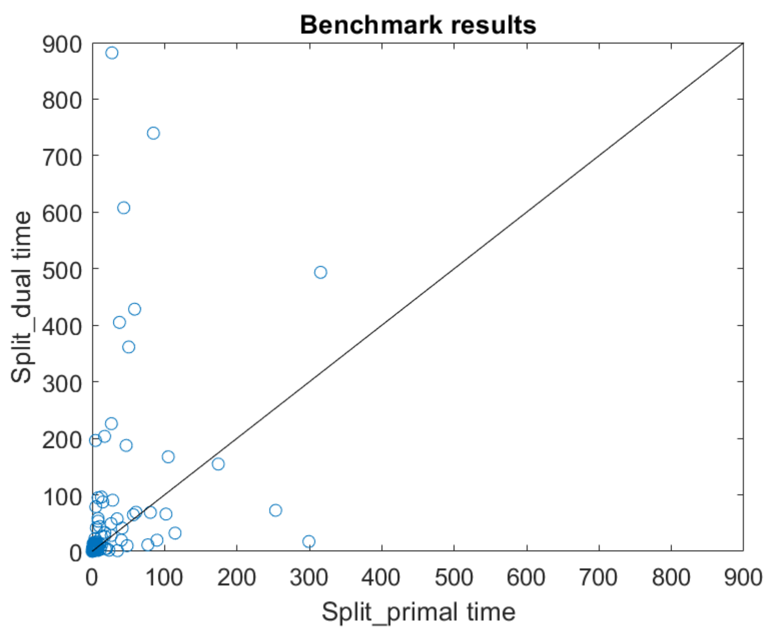


Based on these two graphs, it appears that modified Fukuda's tends to be faster on the harder instances; however, original Fukuda's may still be preferred as it is more reliable.

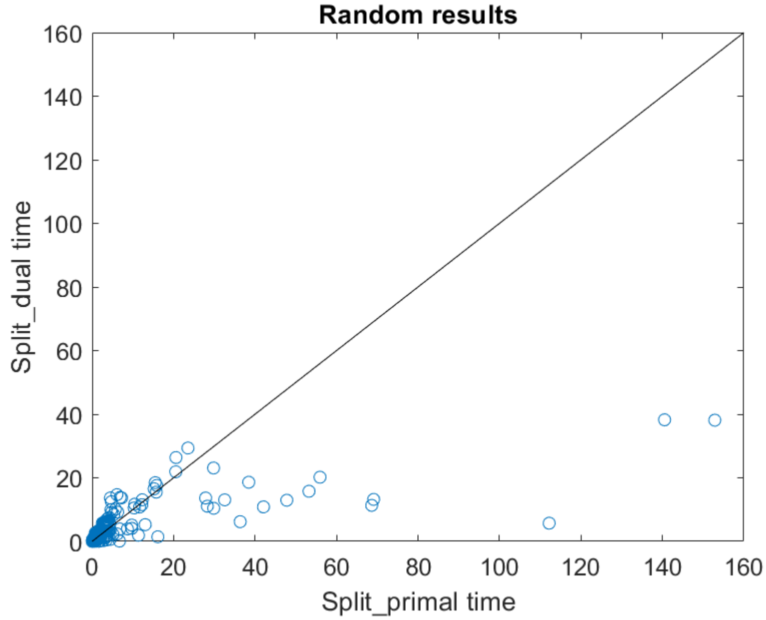
4.4.2 Splitting

Of the 240 benchmark instances, there are five that were not solved by any of the methods within the 15 minute time limit. Referring back to Table 4.1, we see that in addition to these five unsolved instances, the primal splitting method reached the time limit for two other instances and the dual splitting method reached the time limit for 6 other instances. The two that were not solved within the time limit by the primal splitting method were solved by the dual splitting method, and the six that were not solved within the time limit by the dual splitting method were solved by the primal splitting method. In total, 225 of the 240 benchmark instances were solved within the time limit by both splitting methods:

Referring back to Table 4.2, we see that neither splitting method ever reached the time limit on the randomly generated instances; however, the dual splitting method did have some numerical issues. These issues came after the feasibility check confirmed a nonempty feasible region in the form of an infeasibility status. We know the dual splitting LP is not infeasible as **0** is a feasible solution.



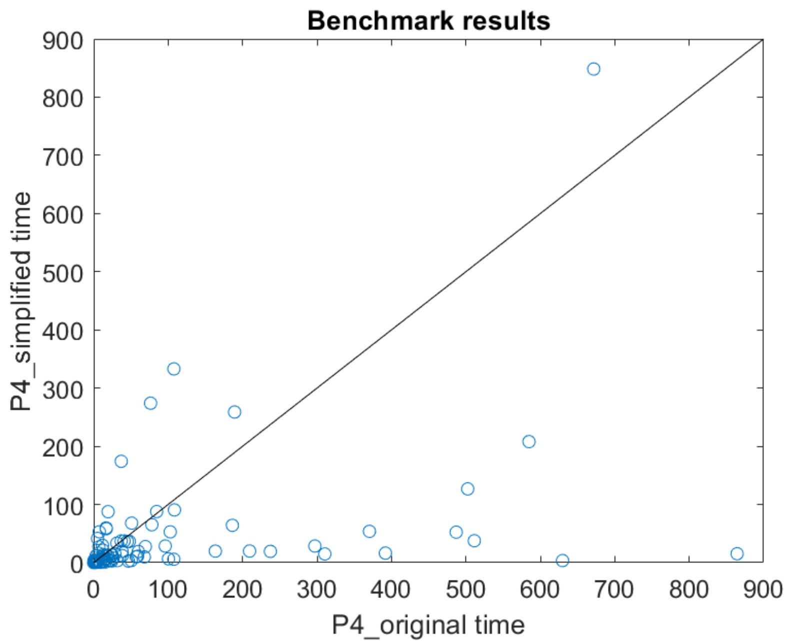
While the two splitting methods seem fairly comparable, the primal splitting methods appears to be slightly more stable based on our numerical experiments. Also, if one is unable to assume feasibility, then the dual splitting method will likely be slower as it necessitates a feasibility check before solving for implicit equalities.



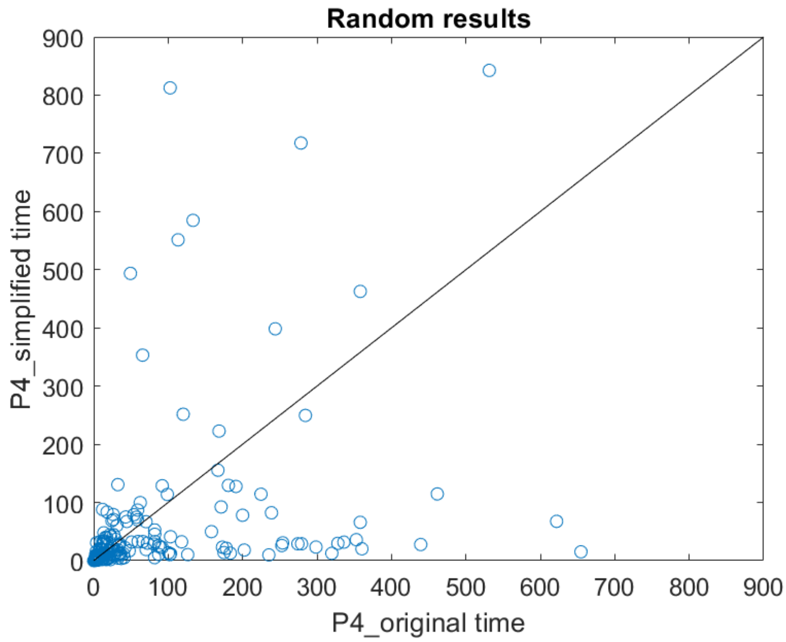
4.4.3 P4

In our code for P4, we use a tolerance of 10^{-5} to determine whether s is positive, negative, or zero. To determine whether or not a dual variable is positive, we use a tolerance of 10^{-8} for P4 and a tolerance of 10^{-6} for our simplified P4.

Referring back to Table 4.1, we see that P4 reached the time limit for just over 25% of the benchmark instances and our simplified P4 reached the time limit for just under 25% of the benchmark instances. Both versions returned an incorrect count for three of the instances. Original P4 had a couple more numerical issues as both versions of P4 had an instance where Gurobi failed to return a termination status, but original P4 also had two instances where Gurobi returned an unbounded status when we know theoretically that P4 is bounded (see Proposition 2). For the remaining 160 instances that each algorithm was able to solve within the time limit, we give the following graph:



Referring back to Table 4.2, we see that both the original P4 and our simplified P4 reached the time limit on about 20 of the randomly generated instances. Original P4 had a more incorrect counts and more numerical issues than our simplified P4. For the numerical issues, original P4 had four instances where gurobi returned a numeric status. Of the incorrect counts, there were two instances that were incorrectly determined to be infeasible by the original P4 method. Similarly, there was one instance that was incorrectly determined to be infeasible by the simplified P4 method. The remainder of the incorrect counts were instances where a nonnegative number was returned, but it was did not match the number of implicit equalities present in the polyhedron. For the 345 instances that each algorithm was able to solve within the time limit, we give the following graph:



4.5 Conclusion

Overall, it seems that the P4 methods are the slowest and that the splitting methods are the fastest and most reliable. We remark that for both Fukuda's methods and P4 methods, tolerance choice is very important. You may note that the tolerances used in our numerical experiments vary depending on the method. For instance, when determining whether or not a dual variable was positive, a tolerance or 10^{-8} was used for the original P4 and a tolerance of 10^{-6} was used for simplified P4. Initially, a tolerance of 10^{-8} was used for both methods, but with this tolerance, simplified P4 returned an incorrect count for more than 100 of the nearly 389 randomly generated instances. Based on preliminary computational results such as these, we adjusted the tolerances until the results were more promising. Since it can be difficult to determine a good tolerance and what is a good tolerance may vary from problem to problem, this is a drawback of Fukuda's methods and the P4 methods. A major reason why the splitting methods are more reliable than the other methods is because they do not rely on tolerances when counting the number of implicit equalities. The variables that the splitting method counts are binary, so it is much easier to determine whether a given variable is positive.

Chapter 5

Conclusions and Discussion

In Chapter 2, we show that the lifted convex hull of the union of two nonempty closed sets is equal to the convex hull of the union of the individual lifted convex hulls. Encouraged by this result, we examine the feasible regions of two variants of the trust-region subproblem. We are able to determine a hyperplane that partitions the feasible region into two nonempty closed subset so that we can write the feasible region as the union of these two sets with known lifted convex hulls. This allows us to obtain an exact semidefinite representable reformulation for each of the two problems of interest, and our computational results show that these reformulations perform well.

In Chapter 3, we consider the unit commitment problem. Using a completely positive reformulation of the original mixed-binary formulation, we are able to achieve a semidefinite programming relaxation of the unit commitment problem. While preliminary computational results support the idea that shadow prices of our SDP may be useful in a pricing mechanism, there is still a need for further tests and fine-tuning. There is also concern that given the large-scale of electricity markets, the SDP formulation may still be challenging to solve efficiently.

In Chapter 4, we review four algorithms for finding implicit equalities, two of which we concluded were equivalent. We provide a modification for each algorithm seeking to improve their performances. After comparing their numerical performance, we conclude that the splitting methods are superior. Going forward, we seek to understand for which problem structures the dual splitting method would perform better than the primal splitting method.

Appendices

Appendix A Simple examples pertaining to SDP reformulations

Consider a general quadratically-constrained quadratic program (QCQP):

$$\begin{aligned} \inf \quad & \mathbf{x}^T H \mathbf{x} + 2\mathbf{g}^T \mathbf{x} \\ \text{s.t.} \quad & \mathbf{x} \in \mathcal{F}, \end{aligned}$$

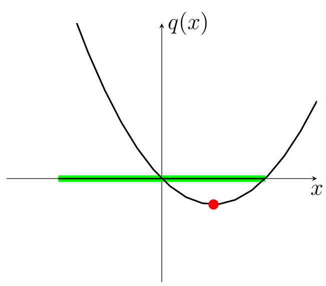
where $H \in \mathcal{S}^n$ and $\mathcal{F} \subseteq \mathbb{R}^n$ is a nonempty closed set. This problem can be equivalently lifted to

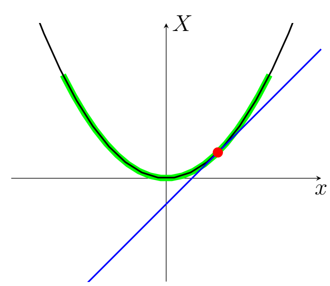
$$\begin{aligned} \inf \quad & H \bullet X + 2\mathbf{g}^T \mathbf{x} \\ \text{s.t.} \quad & X = \mathbf{x}\mathbf{x}^T \\ & \mathbf{x} \in \mathcal{F} \end{aligned} \tag{1}$$

where $(\mathbf{x}, X) \in \mathbb{R}^n \times \mathcal{S}^n$ and $H \bullet X = \mathrm{Tr}(H^T X)$.

Example 2. *Consider the following example. On the left-hand side, we give a QCQP in* $\mathbb{R}$*, and on the right-hand side, we give its lifted reformulation.*

$$\begin{aligned} \min \quad & q(x) := x^2 - x \\ s.t. \quad & x \in \mathcal{F} := \{x | x^2 \leq 1\} \end{aligned} \qquad\qquad \begin{aligned} \min \quad & X - x \\ s.t. \quad & X = x^2 \\ & X \leq 1 \end{aligned}$$





Moreover, it can be shown (e.g. in [43] and [20]) that (1) is equivalent to

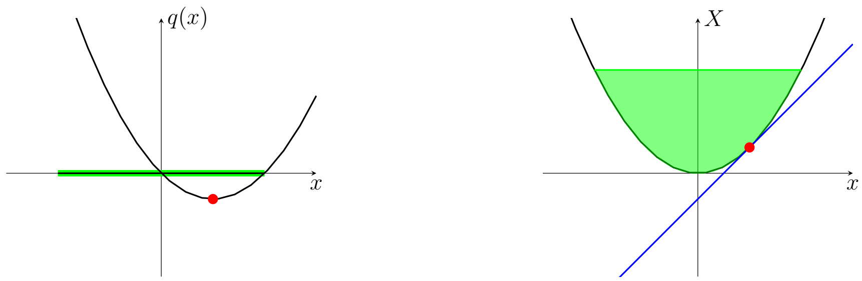
$$\begin{aligned} \inf \quad & H \bullet X + 2\mathbf{g}^T\mathbf{x} \\ \text{s.t.} \quad & (x, X) \in \mathcal{C}(\mathcal{F}), \end{aligned}$$

where $\mathcal{C}(\mathcal{F}) := \text{conv}\{(\mathbf{x}, \mathbf{x}\mathbf{x}^T) \mid \mathbf{x} \in \mathcal{F}\}$ is the lifted convex hull.

Example 3. *Continuing with the QCQP in Example 1, we have an equivalence between*

$$\begin{aligned} \min \quad & q(x) := x^2 - x \\ s.t. \quad & x \in \mathcal{F} := \{x | x^2 \leq 1\} \end{aligned} \qquad \textit{and} \qquad \begin{aligned} \min \quad & X - x \\ s.t. \quad & (x, X) \in \mathcal{C}(\mathcal{F}) \end{aligned}$$

where $\mathcal{C}(\mathcal{F}) = \text{conv}\{(x, x^2) \mid x \in \mathcal{F}\} = \{(x, X) \mid X \geq x^2, X \leq 1\}$.



Bibliography

[1] Temadher A Almaadeed, Saeid Ansary Karbasy, Maziar Salahi, and Abdelouahed Hamdi. On indefinite quadratic optimization over the intersection of balls and linear constraints. *Journal of Optimization Theory and Applications*, 194(1):246–264, 2022.

[2] Saeid Ansary Karbasy and Maziar Salahi. Quadratic optimization with two ball constraints. *Numerical Algebra, Control & Optimization*, 10(2):165, 2020.

[3] Saeid Ansary Karbasy and Maziar Salahi. On the branch and bound algorithm for the extended trust-region subproblem. *Journal of Global Optimization*, 83:221–233, 2022.

[4] Kurt M Anstreicher. Kronecker product constraints with an application to the two-trust-region subproblem. *SIAM Journal on Optimization*, 27(1):368–378, 2017.

[5] Kurt M Anstreicher. Solving two-trust-region subproblems using semidefinite optimization with eigenvector branching. *Journal of Optimization Theory and Applications*, pages 1–17, 2022.

[6] Mosek ApS. Mosek optimization toolbox for matlab. *User's Guide and Reference Manual, Version*, 4, 2019.

[7] Mosek ApS. Mosek modeling cookbook, 2020.

[8] Dennis S Arnon, George E Collins, and Scott McCallum. Cylindrical algebraic decomposition i: The basic algorithm. *SIAM Journal on Computing*, 13(4):865–877, 1984.

[9] Xiaowei Bao, Nikolaos V Sahinidis, and Mohit Tawarmalani. Semidefinite relaxations for quadratically constrained quadratic programming: A review and comparisons. *Mathematical Programming*, 129:129–157, 2011.

[10] Dimitris Bertsimas and Ryan Cory-Wright. On polyhedral and second-order cone decompositions of semidefinite optimization problems. *Operations Research Letters*, 48(1):78–85, 2020.

[11] 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*, pages 380–390. 2014.

[12] Immanuel M Bomze, Vaithilingam Jeyakumar, and Guoyin Li. Extended trust-region problems with one or two balls: exact copositive and lagrangian relaxations. *Journal of Global Optimization*, 71(3):551–569, 2018.

[13] Samuel Burer. On the copositive representation of binary and continuous nonconvex quadratic programs. *Mathematical Programming*, 120(2):479–495, 2009.

[14] Samuel Burer. Copositive programming. *Handbook on semidefinite, conic and polynomial optimization*, pages 201–218, 2012.

[15] Samuel Burer. A gentle, geometric introduction to copositive optimization. *Mathematical Programming*, 151(1):89–116, 2015.

[16] Samuel Burer and Kurt M. Anstreicher. Second-order-cone constraints for extended trust-region subproblems. *SIAM Journal on Optimization*, 23(1):432–451, 2013.

[17] Samuel Burer and Boshi Yang. The trust region subproblem with non-intersecting linear constraints. *Mathematical Programming*, 149(1-2):253–264, 2015.

[18] Miguel Carrión and José M Arroyo. A computationally efficient mixed-integer linear formulation for the thermal unit commitment problem. *IEEE Transactions on Power Systems*, 21(3):1371–1378, 2006.

[19] Dmitriy Drusvyatskiy, Henry Wolkowicz, et al. The many faces of degeneracy in conic optimization. *Foundations and Trends® in Optimization*, 3(2):77–170, 2017.

[20] Gabriele Eichfelder and Janez Povh. On the set-semidefinite representation of nonconvex quadratic programs over arbitrary feasible sets. *Optimization Letters*, 7(6):1373–1386, 2013.

[21] Anders Eltved and Samuel Burer. Strengthened SDP relaxation for an extended trust region subproblem with an application to optimal power flow. *Mathematical Programming*, 197:281–306, 2023.

[22] Salar Fattahi, Morteza Ashraphijuo, Javad Lavaei, and Alper Atamtürk. Conic relaxations of the unit commitment problem. *Energy*, 134:1079–1095, 2017.

[23] Robert Michael Freund, Robin Roundy, and Michael J Todd. Identifying the set of always-active constraints in a system of linear inequalities by a single linear program. 1985.

[24] Komei Fukuda. Lecture: Polyhedral computation, spring 2016. *Institute for Operations Research and Institute of Theoretical Computer Science, ETH Zurich. https://inf. ethz. ch/personal/fukudak/lect/pclect/notes2015/PolyComp2015. pdf*, 2016.

[25] Ambros Gleixner, Gregor Hendel, Gerald Gamrath, Tobias Achterberg, Michael Bastubbe, Timo Berthold, Philipp M. Christophel, Kati Jarck, Thorsten Koch, Jeff Linderoth, Marco Lübbecke, Hans D. Mittelmann, Derya Ozyurt, Ted K. Ralphs, Domenico Salvagnin, and Yuji Shinano. MIPLIB 2017: Data-Driven Compilation of the 6th Mixed-Integer Programming Library. *Mathematical Programming Computation*, 2021.

[26] Nicholas I. M. Gould, Stefano Lucidi, Massimo Roma, and Philippe L. Toint. Solving the trust-region subproblem using the lanczos method. *SIAM Journal on Optimization*, 9(2):504–525, 1999.

[27] Michael Grant and Stephen Boyd. CVX: Matlab software for disciplined convex programming, version 2.2. `http://cvxr.com/cvx`, January 2020.

[28] Paul R Gribik, William W Hogan, Susan L Pope, et al. Market-clearing electricity prices and energy uplift. *Cambridge, MA*, pages 1–46, 2007.

[29] Cheng Guo, Merve Bodur, and Joshua A Taylor. Copositive duality for discrete markets and games. *arXiv preprint arXiv:2101.05379*, 2021.

[30] Vincent Hindriksen. How expensive is an operation on a cpu? `https://streamhpc.com/blog/2012-07-16/how-expensive-is-an-operation-on-a-cpu/`, 2012.

[31] Nam Ho-Nguyen and Fatma Kılınç-Karzan. A second-order cone based approach for solving the trust-region subproblem and its variants. *SIAM Journal on Optimization*, 27(3):1485–1512, 2017.

[32] William W Hogan and Brendan J Ring. On minimum-uplift pricing for electricity markets. *Electricity Policy Group*, pages 1–30, 2003.

[33] Hao Hu and Boshi Yang. Affine fr: an effective facial reduction algorithm for semidefinite.

[34] Robin Broder Hytowitz, Bethany Frew, Gordon Stephen, Erik Ela, Nikita Singhal, Aaron Bloom, and Jessica Lau. Impacts of price formation efforts considering high renewable penetration levels and system resource adequacy targets. Technical report, National Renewable Energy Lab.(NREL), Golden, CO (United States), 2020.

[35] V. Jeyakumar and G.Y. Li. Trust-region problems with linear inequality constraints: exact SDP relaxation, global optimality and robust optimization. *Mathematical Programming*, 147(1-2):171–206, 2014.

[36] Rujun Jiang and Duan Li. Novel reformulations and efficient algorithms for the generalized trust region subproblem. *SIAM Journal on Optimization*, 29(2):1603–1633, 2019.

[37] Qingwei Jin, Ye Tian, Zhibin Deng, Shu-Cherng Fang, and Wenxun Xing. Exact computable representation of some second-order cone constrained quadratic programming problems. *Journal of the Operations Research Society of China*, 1(1):107–134, 2013.

[38] Alexander Joyce and Boshi Yang. Convex hull results on quadratic programs with non-intersecting constraints. *Mathematical Programming*, pages 1–20, 2023.

[39] Sarah Kelly, Yuyuan Ouyang, and Boshi Yang. Semidefinite representable reformulations for two variants of the trust-region subproblem. *Operations Research Letters*, 51(6):695–701, 2023.

[40] Sunyoung Kim and Masakazu Kojima. Exact solutions of some nonconvex quadratic optimization problems via SDP and SOCP relaxations. *Computational Optimization and Applications*, 26:143–154, 2003.

[41] Benoît Legat, Oscar Dowson, Joaquim Dias Garcia, and Miles Lubin. Mathoptinterface: a data structure for mathematical optimization problems. *INFORMS Journal on Computing*, 34(2):672–689, 2022.

[42] George Liberopoulos and Panagiotis Andrianesis. Critical review of pricing schemes in markets with non-convex costs. *Operations Research*, 64(1):17–31, 2016.

[43] Cheng Lu, Shu-Cherng Fang, Qingwei Jin, Zhenbo Wang, and Wenxun Xing. KKT solution and conic relaxation for solving quadratically constrained quadratic programming problems. *SIAM Journal on Optimization*, 21(4):1475–1490, 2011.

[44] Mahmood Mehdiloo. On finding a relative interior point of a polyhedral set. *Mathematics Interdisciplinary Research*, 6(2):121–138, 2021.

[45] Mahmood Mehdiloozad, Kaoru Tone, Rahim Askarpour, and Mohammad Bagher Ahmadi. Finding a maximal element of a non-negative convex set through its characteristic cone: An application to finding a strictly complementary solution. *Computational and Applied Mathematics*, 37(1):53–80, 2018.

[46] Tiago Montanher, Arnold Neumaier, and Ferenc Domes. A computational study of global optimization solvers on two trust region subproblems. *Journal of Global Optimization*, 71(4):915–934, 2018.

[47] Jorge J. Moré and D. C. Sorensen. Computing a trust region step. *SIAM Journal on Scientific and Statistical Computing*, 4(3):553–572, 1983.

[48] Richard P O'Neill, Paul M Sotkiewicz, Benjamin F Hobbs, Michael H Rothkopf, and William R Stewart Jr. Efficient market-clearing prices in markets with nonconvexities. *European Journal of Operational Research*, 164(1):269–285, 2005.

[49] Ting Kei Pong and Henry Wolkowicz. The generalized trust region subproblem. *Computational Optimization and Applications*, 58(2):273–322, 2014.

[50] Edward Quarm and Ramtin Madani. Scalable security-constrained unit commitment under uncertainty via cone programming relaxation. *IEEE Transactions on Power Systems*, 36(5):4733–4744, 2021.

[51] Franz Rendl and Henry Wolkowicz. A semidefinite framework for trust region subproblems with applications to large scale minimization. *Mathematical Programming*, 77(1):273–299, 1997.

[52] James Renegar. On the computational complexity and geometry of the first-order theory of the reals. part i: Introduction. preliminaries. the geometry of semi-algebraic sets. the decision problem for the existential theory of the reals. *Journal of Symbolic Computation*, 13(3):255–299, 1992.

[53] Carlos Ruiz, Antonio J Conejo, and Steven A Gabriel. Pricing non-convexities in an electricity pool. *IEEE Transactions on Power Systems*, 27(3):1334–1342, 2012.

[54] Shinsaku Sakaue, Yuji Nakatsukasa, Akiko Takeda, and Satoru Iwata. Solving generalized cdt problems via two-parameter eigenvalues. *SIAM Journal on Optimization*, 26(3):1669–1694, 2016.

[55] Alexander Schrijver. *Theory of linear and integer programming*. John Wiley & Sons, 1998.

[56] RJ Shanker. Comments on standard market design: Resource adequacy requirement. federal energy regulatory commission. Technical report, Docket RM01-12-000, January 10, 2003.

[57] Jos F. Sturm and Shuzhong Zhang. On cones of nonnegative quadratic functions. *Mathematics of Operations Research*, 28(2):246–267, 2003.

[58] Levent Tunçel. On the slater condition for the sdp relaxations of nonconvex sets. *Operations Research Letters*, 29(4):181–186, 2001.

[59] Alex Wang. *On Quadratically Constrained Quadratic Programs and their Semidefinite Program Relaxations*. PhD thesis, Carnegie Mellon University.

[60] Alex L Wang and Fatma Kılınç-Karzan. The generalized trust region subproblem: solution complexity and convex hull results. *Mathematical Programming*, 191(2):445–486, 2022.

[61] Alex L Wang and Fatma Kılınc-Karzan. On the tightness of SDP relaxations of QCQPs. *Mathematical Programming*, 193(1):33–73, 2022.

[62] Shu Wang and Yong Xia. Strong duality for generalized trust region subproblems: S-lemma with interval bounds. *Optimization Letters*, 9(6):1063–1073, 2015.

[63] Boshi Yang, Kurt Anstreicher, and Samuel Burer. Quadratic programs with hollows. *Mathematical Programming*, 170(2):541–553, 2018.

[64] Boshi Yang and Samuel Burer. A two-variable approach to the two-trust-region subproblem. *SIAM Journal on Optimization*, 26(1):661–680, 2016.

[65] Yinyu Ye and Shuzhong Zhang. New results on quadratic minimization. *SIAM Journal on Optimization*, 14(1):245–267, 2003.