ABSTRACT

DENG, ZHIBIN. Conic Reformulations and Approximations to Some Subclasses of Nonconvex Quadratic Programming Problems. (Under the direction of Dr. Shu-Cherng Fang.)

In this dissertation, some subclasses of nonconvex quadratic programming problems are studied. We first study the nonconvex quadratic programming problem over the standard simplex with application to copositive matrix detection. A sequence of linear conic approximations are derived to bound the original problem by using semidefinite programming techniques. An algorithm based on the adaptive approach is developed to detect the copositivity of a given matrix. Then, we study the nonconvex quadratic programming problem over a set of convex quadratic constraints. A conic reformulation and approximation with an adaptive scheme for this problem is also developed. We proved that an ϵ-optimal solution can be obtained in a finite number of iterations using the proposed algorithm. Finally, we extend our study to the bounded nonconvex quadratically constrained quadratic programming problem. A branch-and-cut algorithm is developed to solve this problem based on some generalized linear and quadratic polar cuts. The finite termination of the proposed algorithm is proved and our numerical results confirm its superior performance over other known algorithms in the literature. Directions for future research are included at the end.

Conic Reformulations and Approximations to Some Subclasses of Nonconvex Quadratic Programming Problems

by
Zhibin Deng

A dissertation submitted to the Graduate Faculty of
North Carolina State University
in partial fulfillment of the
requirements for the Degree of
Doctor of Philosophy

Industrial Engineering

Raleigh, North Carolina

2013

APPROVED BY:

Dr. Russell E. King	Dr. James R. Wilson
Dr. Yunan Liu	Dr. Shu-Cherng Fang Chair of Advisory Committee

DEDICATION

This dissertation is dedicated to my family for their endless love and support:

Yadi Le, my beloved wife

Rainee and Rachel Deng, my sweet daughters

Shuying Cai, my dear Mom

Mingguang Deng, my dear Dad

BIOGRAPHY

I was born in Hongxing, a small town surrounded by hills in Jiangxi, China. I grew up like other children except that I always did exceedingly well at school. In my high school age, I was transferred to No. 1 High School in Dongxiang, where I met my wife. In Fall 2003, I was admitted to the Department of Mathematical Science of Tsinghua University in Beijing. During the next six years, I received my B.S. degree in Applied Mathematics and M.S. degree in Operations Research under the supervision of Dr. Wenxun Xing. In 2009, I joined the North Carolina State University to pursue my Ph.D. degree in the Department of Industrial and Systems Engineering under the supervision of Dr. Shu-Cherng Fang.

ACKNOWLEDGEMENTS

Over the past four and half years I have received support and encouragement from a great number of individuals. Part of this work has been supported by the Edward P. Fitts Fellowship.

I would like to express my deepest and sincerest gratitude to Dr. Shu-Cherng Fang for guiding me through my Ph.D. study. He inspired me not only in how to do researches, but also in how to be a responsible father. He showed me what is the best career path and helped me to achieve the goals. Great thanks also go to Dr. Russell E. King, Dr. James R. Wilson and Dr. Yunan Liu for their valuable comments as my committee members. I am thankful to Dr. Donald E.K. Martin for being the Graduate School Representative in my defense. I am also obliged to Dr. John E. Lavery and Dr. Wenxun Xing for their kindly help in my study.

Thanks to the staff of the Industrial and Systems Engineering Department, especially Ms. Cecilia Chen, Mr. Bill Irwin, Ms. Elaine Erwin and Mr. Justin Lancaster for their support and help. I am also grateful to my fellow friends in the FANGROUP: Pingke Li, Qingwei Jin, Kun Huang, Lan Li, Lu Yu, Yuan Tian, Ye Tian, Chia-Chun Hsu, Ziteng Wang, Jian Luo, Chien-Chia Huang, Tiantian Nie and many others whose names are not listed here. Their friendship and encouragement are always in my mind.

Finally, I am deeply indebted to my wife for always believing in me and supporting me. I could not sustain myself through Ph.D. study without her endless love, patience and understanding.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

Introduction

Quadratically constrained quadratic programming (QCQP) forms an important class of optimization problems. The study of QCQP problems originated from Kuhn and Tucker [71] in 1951 and it is known that QCQP problems are NP-hard in general [85]. The aim of this dissertation is to study the theory of conic reformulations and approximations for solving some important subclasses of QCQP problems. Three fundamental subclasses of QCQP problems are particularly studied. The first one is the quadratic optimization over the standard simplex, the second one is the quadratic optimization over a set of convex quadratic constraints and the third one is the bounded quadratically constrained quadratic programming problem.

1.1 Statement of Problem and Motivation

A quadratically constrained quadratic programming problem can be defined as

$$(\text{QCQP}) \qquad \begin{array}{ll} \min & x^T P^0 x + 2(q^0)^T x + \gamma^0 \\ \text{s.t.} & x^T P^j x + 2(q^j)^T x + \gamma^j \leq 0, \quad j = 1, \ldots, m, \end{array} \tag{1.1}$$

where $P^j \in \mathcal{S}^n$, the space of real symmetric square matrices of order n, $q^j \in \mathbb{R}^n$, the n-dimensional real space, and $\gamma^j \in \mathbb{R}$ for $j = 0, 1, \ldots, m$. Many well-known and hard problems are subclasses of QCQP problems. We list three of our major interests below.

1.1.1 Quadratic Programming Problem over the Standard Simplex

A quadratic programming problem over the standard simplex has the following form:

$$\text{(StQP)} \qquad \begin{aligned} \min \quad & x^T P^0 x \\ \text{s.t.} \quad & e^T x = 1, \; x \geq 0, \end{aligned} \tag{1.2}$$

where $e = (1, \ldots, 1)^T \in \mathbb{R}^n$. This problem is called the standard quadratic programming (StQP) in some literatures [22, 23, 24].

One notable application of problem (StQP) is the detection of copositivity of the matrix P^0. Notice that the matrix P^0 becomes *copositive*, if the optimal value of problem (StQP) is nonnegative. The concept of copositivity can be traced back to Motzkin [80] in 1952. Using the cone of copositive matrices in optimization for reformulating hard problems has been studied only in the last decade. A number of NP-hard problems, such as the binary quadratic problem [34], the fractional quadratic problem [92], determining the clique number of a graph [81], graph partitioning [90] and the quadratic assignment problem [91], have been shown to admit an exact copositive programming reformulation. Unfortunately, Murty and Kabadi [82] proved that detecting a copositive matrix is a co-NP-complete problem in 1987. Consequently, the development of approximation theory and efficient algorithms for detecting whether a given matrix is copositive or not are preliminary requirements for solving these hard problems.

1.1.2 Quadratic Programming Problem over Convex Quadratic Constraints

A quadratic programming problem over a set of convex quadratic constraints has the following form:

$$\text{(ETRS)} \qquad \begin{aligned} \min \quad & x^T P^0 x + 2(q^0)^T x \\ \text{s.t.} \quad & x^T P^j x + 2(q^j)^T x + \gamma^j \leq 0, \quad j = 1, \ldots, m, \end{aligned} \tag{1.3}$$

where $P^0 \in \mathcal{S}^n$ and $P^j \in \mathcal{S}^n_+$, the space of real positive semidefinite matrices of order n, for $j = 1, \ldots, m$. If $m = 1$, P^1 is the identity matrix, $q^1 = 0$ and $\gamma^1 < 0$, then problem (ETRS) becomes the classical trust-region subproblem (TRS), which minimizes a nonconvex quadratic objective function over the unit ball. If $m \geq 2$, then the problem is called the extended trust-region subproblem (ETRS) [36].

Trust-region subproblem is a key subproblem in nonlinear optimization [39] with several efficient algorithms available [53, 79, 95]. Problem (ETRS) carries extra convex constraints in TRS, such as elliptic constraints, parallel cuts and so on. It arises from the analysis and relaxation of NP-hard combinatorial optimization problems [89]. Therefore, a good approximation to problem (ETRS) could help develop efficient estimations for these combinatorial optimization problems. It also provides a better subroutine for solving nonlinear programming problems. However, there is no known polynomial-time algorithm for solving problem (ETRS) in general, even for the case that only one additional strictly convex quadratic constraint is added to TRS.

1.1.3 Bounded Quadratically Constrained Quadratic Programming Problem

A bounded quadratically constrained quadratic programming (BQCQP) problem has the following form:

$$\text{(BQCQP)} \qquad \begin{array}{ll} \min & x^T P^0 x + 2(q^0)^T x \\ \text{s.t.} & x^T P^j x + 2(q^j)^T x + \gamma^j \leq 0, \quad j = 1, \ldots, m, \\ & l \leq x \leq u, \end{array} \tag{1.4}$$

where $P^j \in \mathcal{S}^n$; l, u and $q^j \in \mathbb{R}^n$ for $j = 0, 1, ..., m$; and $\gamma^j \in \mathbb{R}$ for $j = 1, \ldots, m$. Problem (BQCQP) is NP-hard in general because it generalizes many well-known NP-hard problems, such as mixed 0-1 linear programming [101] and bilinear programming [119].

Current methods for solving problem (BQCQP) are branch-and-bound (or branch-and-cut) algorithms with various relaxation schemes embedded [11, 13, 72]. The efficiency of these algorithms is mainly determined by the branch-and-bound (or branch-and-cut) rules and the tightness of the relaxation schemes. Therefore, development of a good estimation and an adaptive branch-and-bound (or branch-and-cut) rule could lead to efficient algorithms for solving problem (BQCQP).

1.2 Approaches and Results

We start from bounding the global optimal value of problem (StQP) by reformulating it as a linear conic programming problem defined on the cone of nonnegative quadratic functions over the standard simplex. This linear conic problem is then approximated by a sequence of linear conic programming problems defined on the cone of nonnegative quadratic functions over a union of ellipsoids. Using linear matrix inequality (LMI) representations, each corresponding problem in the sequence can be solved efficiently by semidefinite programming (SDP) techniques. In order to speed up the convergence of the approximation sequence and to relieve the computational effort for solving linear conic programming problems, an adaptive scheme is adopted to refine the union of the ellipsoids. Based on this scheme, an iterative algorithm is proposed to detect the copositivity of a given matrix. The results for this part of work have been published in [45].

We then extend the results to problem (ETRS). A linear conic programming problem on the cone of nonnegative quadratic functions over the feasible domain of problem (ETRS) can also be introduced and similar approximation cones can be obtained based on a revised adaptive scheme. The approximation cones are further improved by using the reformulation-linearization technique (RLT). If the feasible domain of problem (ETRS) is bounded and has a nonempty interior, our proposed algorithm is shown to be able to find an ϵ-optimal solution in a finite number of iterations for any given small tolerance $\epsilon > 0$. The results for this part of work have been submitted [44].

Finally, we study the problem (BQCQP). The conic reformulation and serval convex relaxations for the problem have been derived. A branch-and-cut algorithm based on linear and quadratic cuts is proposed to solve the problem. It is proven that the proposed algorithm yielded a globally ϵ_r-ϵ_z-optimal solution (with respect to feasibility and optimality, respectively) in a finite number of iterations. In order to enhance the computational speed, an adaptive branch-and-cut rule is developed. The results for this work has been written in a working paper [46].

1.3 Outline of the Dissertation

The rest of the dissertation is organized as follows. In Chapter 2, results on the relations between QCQP problems and linear conic programming problems are reviewed. Matrix decomposition and duality theory of linear conic programming are introduced. In Chapter 3, the problem (StQP) is studied and different approximations and algorithms for detecting copositive matrices are explored. In Chapter 4, the problem (ETRS) is studied and an algorithm based on conic reformulation and approximation is developed. In Chapter 5, the problem (BQCQP) is studied and a branch-and-cut algorithm based on polar cuts is developed. In Chapter 6, we summarize our work and provide some directions for future research.

Chapter 2

Preliminary Knowledge

Pardalos and Vavasis [85] proved that nonconvex quadratic programming problems are in general NP-hard. Therefore, problem (QCQP) defined in (1.1) is an NP-hard problem. In fact, some subcases of QCQP problems including the problem (StQP) defined in (1.2) and problem (ETRS) defined in (1.3) are NP-hard. Since we do not expect to have polynomial time algorithms in the literature for solving the QCQP problems, the existing algorithms for solving problem (QCQP) can be generally divided into two categories:

1. Branch-and-bound algorithms based on the optimality conditions. The Karush-Kuhn-Tucker (KKT) conditions and other global optimality sufficient conditions are applied for designing branch rules while the lower bounds are derived from Lagrangian multipliers or convex relaxations. See [18, 33, 65, 120].

2. Branch-and-bound algorithms based on the convex relaxation and reformulation techniques. There exist various convex relaxation methods for nonconvex quadratic functions. Some good examples are linear programming relaxation, semidefinite programming (SDP) relaxation, second-order cone relaxation and reformulation-linearization techniques (RLT). See [13, 62, 72, 94, 102].

In the rest of this chapter, useful notations, related theory and techniques and major results are introduced for us to study the QCQP problems.

2.1 Notations

In this dissertation, problems are represented by their abbreviations such as (StQP) and (ETRS). For a minimization problem, the feasible domain is defined as the set of feasible solutions whose objective values are strictly less than $+\infty$. For a maximization problem, the feasible domain is defined as the set of feasible solutions whose objective values are strictly greater than $-\infty$. The optimal value of an optimization problem (P) is denoted by $V(P)$.

Given an optimization problem (P), let its feasible domain be $\mathcal{F}_P \subseteq \mathbb{R}^n$. For any $x \in \mathcal{F}_P$, if there exists an open subset of $\mathcal{F}_P$ containing x, then this open set is a neighborhood of x and x is an interior point of $\mathcal{F}_P$. The set of all interior points is called the interior of $\mathcal{F}_P$, denoted by $\mathtt{int}(\mathcal{F}_P)$. The smallest closed set containing $\mathcal{F}_P$ is called the closure of $\mathcal{F}_P$, denoted by $\mathtt{cl}\{\mathcal{F}_P\}$. We have

$$\mathtt{int}(\mathcal{F}_P) \subseteq \mathcal{F}_P \subseteq \mathtt{cl}\{\mathcal{F}_P\} \text{ and } \mathtt{cl}\{\mathtt{int}(\mathcal{F}_P)\} = \mathtt{cl}\{\mathcal{F}_P\}.$$

The notation $\mathcal{S}^n$ denotes the set of real symmetric matrices of order n, $\mathcal{S}^n_+$ denotes the set of positive semidefinite matrices of order n, and $\mathcal{S}^n_{++}$ denotes the set of positive definite matrices of order n. For two real symmetric matrices $A = (A_{ij})$ and $B = (B_{ij})$ in $\mathcal{S}^n$, the inner product of A and B is defined by

$$A \bullet B = \sum_{i=1}^{n} \sum_{j=1}^{n} A_{ij} B_{ij}. \tag{2.1}$$

2.2 QCQP and Linear Conic Programming Problems

A *cone* $\mathcal{K}$ is a subset of a given space that satisfies

$$\lambda x \in \mathcal{K} \quad \text{for all } x \in \mathcal{K} \text{ and } \lambda \geq 0.$$

If $\mathcal{K}$ has the property of

$$\text{“}x \in \mathcal{K} \text{ and } -x \in \mathcal{K}\text{” if and only if “}x = 0\text{”},$$

then it is a *pointed* cone. The cone $\mathcal{K}$ is *solid* if it has a nonempty interior. If $\mathcal{K}$ is pointed, solid, closed and convex, then we say the cone $\mathcal{K}$ is *proper*.

Given a set $\mathcal{F} \subseteq \mathbb{R}^n$, the *convex hull* of $\mathcal{F}$, denoted by $\mathtt{conv}\{\mathcal{F}\}$, is defined as the smallest convex set containing $\mathcal{F}$, and the *conic hull* of $\mathcal{F}$, denoted by $\mathtt{cone}\{\mathcal{F}\}$, is defined as the smallest convex cone containing $\mathcal{F}$. From [28], we know

$$\mathtt{conv}\{\mathcal{F}\} = \left\{ x \in \mathbb{R}^n \;\middle|\; x = \sum_{i=1}^{r} \alpha_i x^i \text{ for some } r \in \mathbb{N},\ x^i \in \mathcal{F},\ 0 \leq \alpha_i \leq 1, \right.$$
$$\left. i = 1, \ldots, r, \text{ such that } \sum_{i=1}^{r} \alpha_i = 1 \right\},$$

and

$$\texttt{cone}\{\mathcal{F}\} = \left\{ x \in \mathbb{R}^n \;\middle|\; x = \sum_{i=1}^{r} \alpha_i x^i \text{ for some } r \in \mathbb{N},\ x^i \in \mathcal{F},\ \alpha_i \geq 0 \text{ and } i = 1, \ldots, r \right\},$$

where $\mathbb{N}$ is the set of positive integers. It is not difficult to see that

$$\texttt{conv}\{\mathcal{F}\} \subseteq \texttt{cone}\{\mathcal{F}\} \text{ and } \texttt{cone}\{\texttt{conv}\{\mathcal{F}\}\} = \texttt{cone}\{\mathcal{F}\}.$$

Given a cone $\mathcal{K}$, a linear conic programming (LCoP) problem is defined as

$$\text{(LCoP)} \qquad \begin{array}{ll} \min & C \cdot X \\ \text{s.t.} & A_i \cdot X = b_i,\ i = 1, \ldots, m, \\ & X \in \mathcal{K} \end{array} \tag{2.2}$$

where the notation "$\cdot$" is the inner product in the relevant space. Among all the linear conic programming problems, three subclasses are widely used in theoretical study and practical computing. They are linear programming (LP) problems, second order cone programming (SOCP) problems and positive semidefinite programming (SDP) problems.

In an LP problem, $\mathcal{K} = \mathbb{R}^n_+$, C, X, A_i, $i = 1, \ldots, m$, are vectors in $\mathbb{R}^n$, $b = (b_1, \ldots, b_m)^T \in \mathbb{R}^m$ and the inner product $\cdot$ is defined by $X \cdot Y = X^T Y$.

In an SOCP problem, $\mathcal{K} = \{[\begin{smallmatrix} t \\ x \end{smallmatrix}] \in \mathbb{R}^{n+1} \big| t^2 \geq x^T x \text{ for } t \in \mathbb{R}_+ \text{ and } x \in \mathbb{R}^n\} \subseteq \mathbb{R}^{n+1}$ is the second order cone, C, X, A_i, $i = 1, \ldots, m$, are vectors in $\mathbb{R}^{n+1}$, $b = (b_1, \ldots, b_m)^T \in \mathbb{R}^m$ and the inner product $\cdot$ is defined by $X \cdot Y = X^T Y$.

In an SDP problem, $\mathcal{K} = \mathcal{S}^n_+$, C, X, A_i, $i = 1, \ldots, m$, are matrices in $\mathcal{S}^n$, $b = (b_1, \ldots, b_m)^T \in \mathbb{R}^m$ and the inner product $\cdot$ is defined by $X \cdot Y = X \bullet Y$ as defined in (2.1).
All these three types of problems can be solved in polynomial time (ref. [28, 48, 124]).

In order to derive the dual problem of (LCoP), we need the concept of dual cone. The dual set of a nonempty set $\mathcal{F}$ is defined as

$$\mathcal{F}^* = \left\{ x \;\middle|\; x \cdot y \geq 0 \text{ for all } y \in \mathcal{F} \right\}. \tag{2.3}$$

Note that $\mathcal{F}^*$ is always a closed and convex cone. When $\mathcal{F} = \mathcal{K}$ is a cone, the dual cone $\mathcal{K}^*$ of $\mathcal{K}$ is defined as

$$\mathcal{K}^* = \left\{ x \;\middle|\; x \cdot y \geq 0 \text{ for all } y \in \mathcal{K} \right\}. \tag{2.4}$$

Dual cones satisfy the following properties:

- $\mathcal{K}^*$ is closed and convex.
- $\mathcal{K}_1 \subseteq \mathcal{K}_2$ implies $\mathcal{K}_2^* \subseteq \mathcal{K}_1^*$.

- If $\mathcal{K}$ is solid, then $\mathcal{K}^*$ is pointed.
- If the closure of $\mathcal{K}$ is pointed, then $\mathcal{K}^*$ is solid.
- $(\mathcal{K}^*)^*$ is the closure of the convex hull of $\mathcal{K}$.

These properties show that if $\mathcal{K}$ is a proper cone, then so is its dual $\mathcal{K}^*$ and $(\mathcal{K}^*)^* = \mathcal{K}$.

By using the concept of dual cone, the dual problem of (LCoP) has the following form

$$\text{(LCoD)} \qquad \begin{array}{ll} \max & b^T y \\ \text{s.t.} & C - \sum_{i=1}^m y_i A_i \in \mathcal{K}^* \end{array} \tag{2.5}$$

Since the proper cones have more desirable properties, without any specific statement, we assume that $\mathcal{K}$ and $\mathcal{K}^*$ are proper in problems (LCoP) and (LCoD), respectively, in the rest of this chapter.

Sturm and Zhang [117] established the equivalence relation between quadratic programming problems and linear conic programming problems. In fact, any quadratic optimization problem has an equivalent linear conic programming problem form. In order to establish this equivalence relation, we need two concepts: *homogenization* and the *cone of nonnegative quadratic functions*. Formally, for a nonempty set $\mathcal{F} \subseteq \mathbb{R}^n$, its homogenization is given by

$$\mathscr{H}_{\mathcal{F}} := \mathtt{cl}\left\{ \begin{bmatrix} t \\ x \end{bmatrix} \in \mathbb{R}_{++} \times \mathbb{R}^n \middle| x/t \in \mathcal{F} \right\}, \tag{2.6}$$

which is a closed cone (not necessary to be convex though) in $\mathbb{R}^{n+1}$. The cone of nonnegative quadratic functions over $\mathcal{F}$ is given by

$$\mathscr{N}_{\mathcal{F}} := \left\{ \begin{bmatrix} z_0 & z^T \\ z & Z \end{bmatrix} \in \mathcal{S}^{n+1} \middle| x^T Z x + 2 z^T x + z_0 \geq 0 \text{ for all } x \in \mathcal{F} \right\}, \tag{2.7}$$

where $Z \in \mathcal{S}^n$, $z \in \mathbb{R}^n$ and $z_0 \in \mathbb{R}$. It was proved by Sturm and Zhang [117] that the cone $\mathscr{N}_{\mathcal{F}}$ can be represented by vectors from $\mathscr{H}_{\mathcal{F}}$. We state the result in the next theorem.

Theorem 2.2.1. *For any nonempty set $\mathcal{F}$, it holds that*

$$\mathscr{N}_{\mathcal{F}} = \left(\mathtt{conv}\left\{ yy^T \mid y \in \mathscr{H}_{\mathcal{F}} \right\}\right)^*. \tag{2.8}$$

By using the fact that

$$\mathtt{cl}\{\mathtt{cone}\{yy^T | y \in \mathcal{F}\}\} = \mathtt{cone}\{yy^T | y \in \mathtt{cl}\{\mathcal{F}\}\} \tag{2.9}$$

(see Lemma 3.1 in [73] or Lemma 1 in [117]) and $\mathscr{H}_{\mathcal{F}}$ is a closed cone, we can dualize Theorem 2.2.1 to get

$$\mathscr{N}_{\mathcal{F}}^* = \texttt{conv}\{yy^T | y \in \mathscr{H}_{\mathcal{F}}\}. \tag{2.10}$$

Especially, when $\mathcal{F}$ is closed and bounded, we have

$$\mathscr{N}_{\mathcal{F}}^* = \texttt{cone}\left\{ yy^T \middle| y = \begin{bmatrix} 1 \\ x \end{bmatrix}, x \in \mathcal{F} \right\}. \tag{2.11}$$

For the following nonconvex quadratic programming problem (NQP):

$$\text{(NQP)} \qquad \begin{array}{ll} \inf & x^T P^0 x + 2(q^0)^T x + \gamma^0 \\ \text{s.t.} & x \in \mathcal{F}, \end{array} \tag{2.12}$$

where $\emptyset \neq \mathcal{F} \subseteq \mathbb{R}^n$ is a possibly nonconvex domain, it is equivalent to the linear conic programming problem (MP) defined as

$$\text{(MP)} \qquad \begin{array}{ll} \inf & \begin{bmatrix} \gamma^0 & (q^0)^T \\ q^0 & P^0 \end{bmatrix} \bullet Z \\ \text{s.t.} & Z_{11} = 1, \ Z \in \mathscr{N}_{\mathcal{F}}^*. \end{array} \tag{2.13}$$

In principle, the nonconvex quadratic problem (NQP) and the convex problem (MP) are equivalent. But the fact that we can reformulate a general nonconvex problem (NQP) into a linear conic problem (MP) does not necessarily make such a problem easier to solve. In fact, all the implicit “difficult” constraints originating from the feasible domain $\mathcal{F}$ are packed into the cone $\mathscr{N}_{\mathcal{F}}^*$. Only if we can efficiently solve problem (MP) and decompose the optimal solution of (MP) to get a solution of problem (NQP) in polynomial time, then we can say that there exists an efficient algorithm for solving the original problem (NQP).

2.2.1 Some Examples

In this subsection, two cones of nonnegative quadratic functions are studied. The first one is the well-known positive semidefinite cone.

Theorem 2.2.2. $\mathscr{N}_{\mathbb{R}}^n = \mathcal{S}_+^{n+1} = \mathscr{N}_{\mathbb{R}^n}^*$.

Proof. It is easy to see that $\mathcal{S}_+^{n+1} \subseteq \mathscr{N}_{\mathbb{R}^n}$. On the other hand, if there is a nonzero matrix $U = \begin{bmatrix} U_{11} & u^T \\ u & \bar{U} \end{bmatrix} \in \mathscr{N}_{\mathbb{R}^n} \setminus \mathcal{S}_+^{n+1}$ where $U_{11} \in \mathbb{R}$, $u \in \mathbb{R}^n$, and $\bar{U} \in \mathcal{S}_+^n$, then there exists a nonzero vector $y = \begin{bmatrix} t \\ x \end{bmatrix} \in \mathbb{R}^{n+1}$ with $t \in \mathbb{R}$ and $x \in \mathbb{R}^n$ such that $y^T U y < 0$.

If $t \neq 0$, then $y^T U y = t^2 \begin{bmatrix} 1 \\ x/t \end{bmatrix}^T U \begin{bmatrix} 1 \\ x/t \end{bmatrix} < 0$. This contradicts the assumption that $U \in \mathcal{N}_{\mathcal{F}}$. If $t = 0$, then $y^T U y = x^T \bar{U} x < 0$. Considering the vector $\lambda x \in \mathbb{R}^n$ with $\lambda > 0$ being sufficiently large, we have

$$\begin{bmatrix} 1 \\ \lambda x \end{bmatrix}^T U \begin{bmatrix} 1 \\ \lambda x \end{bmatrix} = U_{11} + 2\lambda u^T x + \lambda^2 x^T \bar{U} x < 0$$

where the last inequality holds because $x^T \bar{U} x < 0$ and λ is sufficiently large. This contradicts the assumption $U \in \mathcal{N}_{\mathbb{R}^n}$. Therefore, $\mathcal{N}_{\mathbb{R}^n} = \mathcal{S}_+^{n+1}$. Moreover, by using the fact that $(\mathcal{S}_+^{n+1})^* = \mathcal{S}_+^{n+1}$, we have $\mathcal{N}_{\mathbb{R}^n}^* = (\mathcal{S}_+^{n+1})^* = \mathcal{S}_+^{n+1}$. □

See Figure 2.1 for the plot of $\mathcal{S}_+^2$.

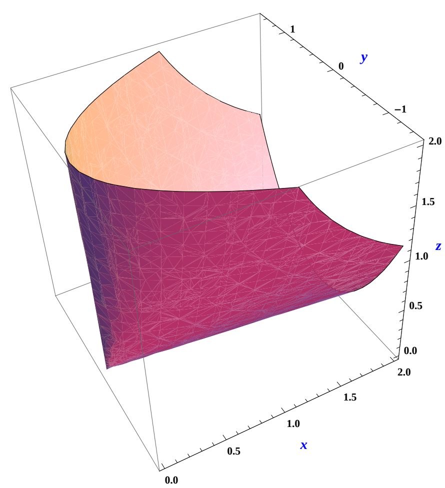

Figure 2.1: Boundary of the positive semidefinite cone $\mathcal{S}_+^2 = \begin{bmatrix} x & y \\ y & z \end{bmatrix}$.

The second one is the *cone of copositive matrices*

$$\mathcal{C}_n = \left\{ M \in \mathcal{S}^n \middle| x^T M x \geq 0 \text{ for all } x \in \mathbb{R}_+^n \right\}. \tag{2.14}$$

Its dual cone, the *completely positive cone*, is defined as

$$\mathcal{C}_n^* = \left\{ M \in \mathcal{S}^n \middle| M = \sum_{i=1}^{r} x^i (x^i)^T \text{ for some } r \in \mathbb{N},\ x^i \in \mathbb{R}_+^n \text{ and } i = 1, \ldots, r \right\}. \tag{2.15}$$

Notice that $\mathcal{C}_n^* \subsetneq \mathcal{S}_+^n \subsetneq \mathcal{C}_n$.

Theorem 2.2.3. *$\mathscr{N}_{\mathbb{R}_+^n} = \mathcal{C}_{n+1}$ and $\mathscr{N}_{\mathbb{R}_+^n}^* = \mathcal{C}_{n+1}^*$.*

Proof. For any $x \in \mathbb{R}_+^n$ and $U \in \mathcal{C}_{n+1}$, $\begin{bmatrix} 1 \\ x \end{bmatrix}^T U \begin{bmatrix} 1 \\ x \end{bmatrix} \geq 0$ holds by the definition of $\mathcal{C}_{n+1}$. Thus, $\mathcal{C}_{n+1} \subseteq \mathscr{N}_{\mathbb{R}_+^n}$. On the other hand, the same argument in Theorem 2.2.2 applies here except that the vector y is in $\mathbb{R}_+^{n+1}$ instead of in $\mathbb{R}^{n+1}$. This proves that $\mathscr{N}_{\mathbb{R}_+^n} = \mathcal{C}_{n+1}$. By dualizing both $\mathscr{N}_{\mathbb{R}_+^n}$ and $\mathcal{C}_{n+1}$, we have $\mathscr{N}_{\mathbb{R}_+^n}^* = \mathcal{C}_{n+1}^*$. $\square$

See Figure 2.2 for the plot of the boundary of $\mathcal{C}_2$.

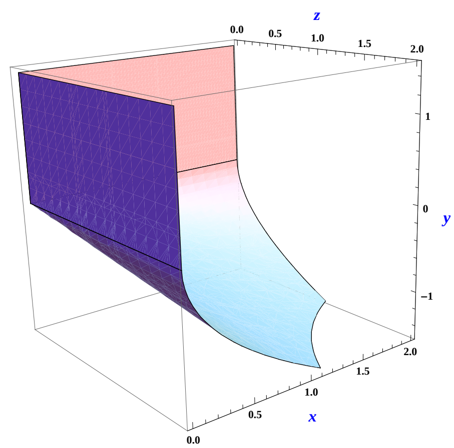

Figure 2.2: Boundary of the copositive cone $\mathcal{C}_2 = \begin{bmatrix} x & y \\ y & z \end{bmatrix}$.

2.3 Matrix Decomposition

Since an efficient algorithm for decomposing a matrix into desired vectors is important in obtaining the optimal solution of problem (NQP), we review some results about matrix decomposition in this section. Related results can be found in [4] and [125].

For any $X \in \mathcal{S}_+^n$, X has a *rank-one decomposition*, that is

$$X = \sum_{i=1}^{r} x^i (x^i)^T$$

where $r \in \mathbb{N}$ is the rank of X and $x^i \in \mathbb{R}^n$ for $i = 1, \ldots, r$ (ref. [52] or [61]). Although the decomposition exists without any doubt for each $X \in \mathcal{S}^n_+$, this decomposition may not satisfy additional conditions, such as $(x^i)^T Y x^i \leq 0$, $i = 1, ..., r$, for a given matrix $Y \in \mathcal{S}^n$. The following theorem says that such decomposition does exist and it can be accomplished in polynomial time. This result is referred to [125].

Theorem 2.3.1. *Let Y be a given symmetric matrix in $\mathcal{S}^n$ and X be a positive semidefinite matrix with rank r, $0 < r \leq n$. Suppose that $X \bullet Y \leq 0$, then there exists a rank-one decomposition of X running in polynomial time to find $x^i \in \mathbb{R}^n$, $i = 1, \ldots, r$, such that*

$$\begin{array}{l} X = \sum_{i=1}^r x^i (x^i)^T, \\ (x^i)^T Y x^i \leq 0, \;\; i = 1, ..., r. \end{array} \tag{2.16}$$

In particular, one can always find x^i, $i = 1, \ldots, r$, in polynomial time such that

$$\begin{array}{l} X = \sum_{i=1}^r x^i (x^i)^T, \\ (x^i)^T Y x^i = X \bullet Y / r, \;\; i = 1..., r. \end{array} \tag{2.17}$$

The proof of Theorem 2.3.1 in [125] is a constructive one, which indicates that the decomposition can be achieved in polynomial time. The following theorem from [4] deals with a more complicated case in which two equality constraints need to be satisfied.

Theorem 2.3.2. *Let Y_1, Y_2 be any two symmetric matrices in $\mathcal{S}^n$ and $X = x^1(x^1)^T + \cdots + x^r(x^r)^T$ with $3 \leq r \leq n$. If there exist δ_1 and δ_2 satisfying*

$$\begin{array}{l} (x^1)^T Y_1 x^1 = (x^2)^T Y_2 x^2 = \delta_1, \\ ((x^1)^T Y_2 x^1 - \delta_2)((x^2)^T Y_1 x^2 - \delta_2) < 0 \end{array} \tag{2.18}$$

then one can find a vector $\tilde{x}^1 \in \mathbb{R}^n$ in polynomial time such that $X = \tilde{x}^1(\tilde{x}^1)^T + \cdots + \tilde{x}^r(\tilde{x}^r)^T$ and

$$\begin{array}{l} (\tilde{x}^1)^T Y_1 \tilde{x}^1 = \delta_1, \\ (\tilde{x}^1)^T Y_2 \tilde{x}^1 = \delta_2. \end{array} \tag{2.19}$$

The results of matrix decomposition in a complex vector space can be found in [5, 63, 75]. Since our interest is in the real space, those results are omitted here. Recent results on matrix decomposition can be found in [66] and [118]. In general, the rank-one decomposition of a matrix with additional constraints to be satisfied is not an easy job.

2.4 Duality Theory of LCoP Problems

Problem (QCQP) is a special case of problem (NQP), hence it can also be reformulated as a linear conic programming problem, which has the same form as (MP) defined in (2.13). Consequently, the optimality conditions and duality properties of (LCoP) are useful in solving problem (QCQP). In this section, we review the optimality conditions and duality theorems of linear conic programming problems. For the optimality conditions and duality properties on nonlinear conic programming problems, one may refer to [6, 7, 40, 86, 112, 113, 114, 115].

First, we introduce the weak duality theorem for problem (LCoP).

Theorem 2.4.1 (Weak Conic Duality Theorem)**.** *Assume that problems* (LCoP) *and* (LCoD) *are both feasible. Then, the optimal value of problem* (LCoD) *is a lower bound for the optimal value of problem* (LCoP)*.*

The weak duality theorem for problem (LCoP) is much weaker than the duality theorem for linear programming (LP). For LP problems, as long as both primal and dual problems are feasible, then we actually have the strong duality property, *i.e.*, the optimal values of the primal and dual problems are equal. However, in order to get a similar strong duality result for problem (LCoP), we require the condition that the primal problem (LCoP) is *strictly feasible*, i.e., there exists an $X \in \mathtt{int}(\mathcal{K})$ such that $A_i \cdot X = b_i$ for $i = 1, \ldots, m$. Geometrically speaking, $\mathcal{A} \cap \mathtt{int}(\mathcal{K}) \neq \emptyset$, where $\mathcal{A} = \{X \big| A_i \cdot X = b_i,\ i = 1, \ldots, m\}$ is an affine space. Similarly, we say problem (LCoD) is strictly feasible if there exists $y = (y_1, \ldots, y_m)^T$ such that $C - \sum_{i=1}^m y_i A_i \in \mathtt{int}(\mathcal{K}^*)$. Then we have the next strong duality theorem.

Theorem 2.4.2 (Strong Conic Duality Theorem)**.** *Consider problem* (LCoP) *defined in* (2.2) *along with its conic dual problem* (LCoD) *defined in* (2.5)*.*

a. If problem (LCoP) *is bounded below and strictly feasible, then problem* (LCoD) *is feasible, an optimal solution is attainable for problem (LCoD) and the optimal values of problems* (LCoP) *and* (LCoD) *are equal.*

b. If problem (LCoD) *is bounded above and strictly feasible, then problem* (LCoP) *is feasible, an optimal solution is attainable for problem (LCoP) and the optimal values of problems* (LCoP) *and* (LCoD) *are equal.*

Based on the strong conic duality theorem, the conic optimality conditions for a primal-dual feasible pair $(X^*,\ y^*)$ can be derived. The result is similar to the optimality conditions for LP problems.

Corollary 2.4.3 (Conic Optimality Conditions)**.** *Assume that at least one of the problems* (LCoP) *and* (LCoD) *is bounded and strictly feasible. Then a primal-dual feasible pair* $(X^*,\ y^*)$

is a pair of optimal solutions to the respective problems if and only if

$$C \cdot X^* = b^T y^* \qquad \text{(zero duality gap)}$$

or

$$X^* \cdot \left(C - \sum_{i=1}^{m} y_i^* A_i \right) = 0 \qquad \text{(complementary slackness)}$$

For more results on the conic duality theories, one may refer to [20] and [28].

2.5 Linear Matrix Inequality and Reformulation-Linearization Technique

In this section, we introduce two important tools that will be used in the rest of this dissertation.

2.5.1 Linear Matrix Inequality

A *linear matrix inequality* (LMI) is an expression of the form

$$A_0 + y_1 A_1 + \cdots + y_m A_m \succcurlyeq \mathbf{0} \tag{2.20}$$

where $A_0, \ldots, A_m$ are given $n \times n$ symmetric matrices, $y = (y_1, \ldots, y_m)$ is a vector of real variables, and $\succcurlyeq$ is an order on $\mathcal{S}_+^n$, *i.e.*, $B \succcurlyeq \mathbf{0}$ means B is a positive semidefinite matrix. The history of LMIs can go back to 1890 when Lyapunov published his seminal work introducing what we now call Lyapunov theory. In 1940s, Lur'e et al. applied LMIs to important (and difficult) practical problems in control engineering [76]. But only small size LMIs can be solved "by hand." In 1960s, Yakubovich et al. showed how to solve a certain family of LMIs by graphical methods [121, 122]. In late 1980s, Nesterov and Nemirovskii developed interior-point algorithms for solving LMIs [84]. Several interior-point algorithms have been implemented and tested on specific families of LMIs that arise in control theory, and found to be extremely efficient. For the more detailed history of LMIs, one may refer to [27].

There are two reasons to study LMIs in this dissertation:

(i) The form of an LMI is very general. Linear inequalities, convex quadratic inequalities, matrix norm inequalities and various other inequalities can all be rewritten as LMIs [119]. This is very useful when deriving the conic reformulations of QCQP problems. For example, an elliptic constraint described by

$$(x - x_c)^T Q (x - x_c) \leq 1$$

where $Q \in \mathcal{S}^n_{++}$ and $x_c \in \mathbb{R}^n$ can be expressed by the following LMI using the Schur complement lemma [28]:

$$\begin{bmatrix} 1 & (x - x_c)^T \\ (x - x_c) & Q \end{bmatrix} \succcurlyeq \mathbf{0}.$$

(ii) An LMI is a convex constraint and can be solved efficiently [50]. Consequently, additional LMI constraints in convex optimization problems (such as semidefinite programming problems) will not increase the complexity. This is very useful when developing conic approximations to QCQP problems. By adding some LMI constraints to an SDP problem used to estimate the original QCQP problem, we may obtain a better bound. Here, LMI constraints actually play a role of valid inequalities for solving SDP problems.

2.5.2 Reformulation-Linearization Technique

In this subsection, we describe a technique, called *reformulation-linearization technique* (RLT), which generates LMIs for SDP problems. A recent review paper on RLT can be referred to [108].

RLT originated in [1, 2, 3]. It initially focused on solving 0-1 and mixed 0-1 linear and polynomial programming problems [100, 101] and later branched into the more general family of continuous, nonconvex polynomial programming problems [103, 104, 107]. The RLT essentially consists of two steps: a reformulation step in which certain additional nonlinear valid inequalities are automatically generated, and a linearization step in which each product term is replaced by a single continuous variable. Here is an example to demonstrate the procedure of RLT.

Example 2.5.1. Consider the following box constrained nonconvex quadratic programming problem:

$$\text{(BQP)} \qquad \begin{array}{ll} \min & \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}^T \begin{bmatrix} -25 & -1500 & 858 \\ -1500 & -1 & -14 \\ 858 & -14 & -51 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 3112 \\ -4 \\ 162 \end{bmatrix}^T \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \\ \text{s.t.} & 0 \le x_1 \le 1,\ 0 \le x_2 \le 1,\ 0 \le x_3 \le 1. \end{array}$$

The optimal value of problem (BQP) is -2.5 with an optimal solution $x_1 = 0, x_2 = 1, x_3 = 0$.

The SDP relaxation for problem (BQP), due to Shor [110], has the following form:

$$
\text{(BQP-SDP)} \quad
\begin{array}{ll}
\min & \begin{bmatrix} 1 & 1556 & -2 & 81 \\ 1556 & -25 & -1500 & 858 \\ -2 & -1500 & -1 & -14 \\ 81 & 858 & -14 & -51 \end{bmatrix} \bullet \begin{bmatrix} 1 & x_1 & x_2 & x_3 \\ x_1 & X_{11} & X_{12} & X_{13} \\ x_2 & X_{21} & X_{22} & X_{23} \\ x_3 & X_{31} & X_{32} & X_{33} \end{bmatrix} \\
\text{s.t.} & 0 \leq x_1 \leq 1,\ 0 \leq x_2 \leq 1,\ 0 \leq x_3 \leq 1, \\
& \begin{bmatrix} 1 & x_1 & x_2 & x_3 \\ x_1 & X_{11} & X_{12} & X_{13} \\ x_2 & X_{21} & X_{22} & X_{23} \\ x_3 & X_{31} & X_{32} & X_{33} \end{bmatrix} \succcurlyeq \mathbf{0}.
\end{array}
$$

The optimal value of problem (BQP-SDP) is -179.0641, which is a lower bound for problem (BQP). The reformulation step in RLT generates the following LMIs (or valid nonlinear inequalities):

$$x_i(x_j - 1) \leq 0, \ \ i, j \in \{1, 2, 3\}$$

from the constraints $0 \leq x_i,\ x_j \leq 1$ for $i,\ j = 1, 2, 3$. The linearization step in RLT replaces the product terms $x_i x_j$ by a single variable X_{ij} for $i, j = 1, 2, 3$, leading to the linear inequalities:

$$X_{ij} - x_i \leq 0, \ \ i, j \in \{1, 2, 3\}.$$

Therefore, after applying RLT, we arrive at a new relaxation problem:

$$
\text{(BQP-RLT)} \quad
\begin{array}{ll}
\min & \begin{bmatrix} 1 & 1579 & -109 & -40 \\ 1579 & -103 & -1506 & -17 \\ -109 & -1506 & 151 & 27 \\ -40 & -17 & 27 & -48 \end{bmatrix} \bullet \begin{bmatrix} 1 & x_1 & x_2 & x_3 \\ x_1 & X_{11} & X_{12} & X_{13} \\ x_2 & X_{21} & X_{22} & X_{23} \\ x_3 & X_{31} & X_{32} & X_{33} \end{bmatrix} \\
\text{s.t.} & 0 \leq x_1 \leq 1,\ 0 \leq x_2 \leq 1,\ 0 \leq x_3 \leq 1, \\
& \begin{bmatrix} 1 & x_1 & x_2 & x_3 \\ x_1 & X_{11} & X_{12} & X_{13} \\ x_2 & X_{21} & X_{22} & X_{23} \\ x_3 & X_{31} & X_{32} & X_{33} \end{bmatrix} \succcurlyeq \mathbf{0}, \\
& X_{11} - x_1 \leq 0,\ X_{22} - x_2 \leq 0,\ X_{33} - x_3 \leq 0, \\
& X_{12} - x_1 \leq 0,\ X_{12} - x_2 \leq 0,\ X_{13} - x_3 \leq 0, \\
& X_{13} - x_1 \leq 0,\ X_{23} - x_2 \leq 0,\ X_{23} - x_3 \leq 0.
\end{array}
$$

The optimal value of problem (BQP-RLT) is -2.5 with an optimal solution $x_1 = 0, x_2 = 1, x_3 = 0$, which is the same as problem (BQP). Notice that problem (BQP) is NP-hard in general, while problem (BQP-RLT) is polynomial-time solvable.

As we can see from Example 2.5.1, adding LMIs (or valid nonlinear inequalities) generated by RLT could effectively improve the bounds obtained by SDP relaxation. This technique will be used in our study.

Chapter 3

Quadratic Programming Problems over the Standard Simplex

In this chapter, we focus on quadratic optimization problems over the standard simplex and its application to the detection of copositive matrices. A sequence of linear conic programming problems are solved to approximate the original problem by using the semidefinite programming techniques. An adaptive approximation scheme is developed to speed up the convergence and relieve the computational effort of the proposed algorithm. Numerical examples and computational results are reported at the end of this chapter.

3.1 Introduction

A real $n \times n$ matrix M is copositive if the homogeneous quadratic form $x^T M x \geq 0$ for all $x \in \mathbb{R}^n_+ = \{x \in \mathbb{R}^n | x \geq 0\}$. Since the copositivity of a nonsymmetric matrix M can be determined by detecting a corresponding symmetric matrix $\frac{M+M^T}{2}$, we assume that the given matrix M is symmetric in this chapter. The copositive cone $\mathcal{C}_n$ is a cone consisting of all $n \times n$ copositive matrices, and obviously $\mathcal{S}^n_+ \subseteq \mathcal{C}_n$. Recall that the inner product of two matrices A, $B \in \mathcal{S}^n$ is defined as $A \bullet B = \mathrm{tr}(A^T B) = \sum_{i=1}^n \sum_{j=1}^n A_{ij} B_{ij}$. The dual cone of $\mathcal{C}_n$, denoted by $\mathcal{C}^*_n \subseteq \mathcal{S}^n$, is the so-called completely positive cone.

The study of copositivity can be traced back to Motzkin [80] in 1952. Both of the cones $\mathcal{C}_n$ and $\mathcal{C}^*_n$ are found to be useful in quadratic and combinatorial optimization. For example, Quist et al. [93] suggested that a stronger convex relaxation for a general quadratic programming problem can be derived by using the copositive cone $\mathcal{C}_n$ rather than the positive semidefinite cone $\mathcal{S}^n_+$. Also, the well-known binary quadratic programming problem can be reformulated as a completely positive programming problem in [34]. Other applications of copositive and completely positive cones in quadratic optimization can be found in [22], [92] and the references therein. As to the applications to combinatorial optimization, de Klerk and Pasechnik [69] derived a copositive formulation for determining the stability number of a graph. Gvozdenović and Laurent found a copositive formulation for the chromatic number of a graph in [56]. Other references include [47] and [90].

Checking whether a given matrix is copositive is proven to be co-NP-complete by Murty and Kabadi in [82]. For a matrix with a special structure, such as being tridiagonal and acyclic, checking its copositivity is possible in linear time ([21] and [64]). For symmetric matrices of order no more than 5, Andersson et al. [10] gave the necessary and sufficient conditions to determine copositivity. Hiriart-Urruty and Seeger [59] wrote a review on copositivity criteria based on matrix structural properties. The above approaches are only suitable for detecting the copositivity of moderate size matrices because of computational requirements. For this purpose, Bundfuss and Dür [31] proposed using global optimization techniques to check copositivity. Their criterion arises from the representation of the quadratic form in barycentric coordinates with respect to the standard simplex and its simplicial partitions thereof. As the partition gets finer and finer, all strictly copositive matrices are captured. This approach gives very good numerical results for many matrices. Most recently, Bomze and Eichfelder [25] present three new copositivity tests based on the difference-of-convex (d.c.) decompositions, and incorporate them into a branch-and-bound algorithm of the ω-subdivision type. These tests employ linear programming and convex quadratic programming techniques. The results of their numerical experiments look very promising. Related papers include [42] and [67]. Other works worth mentioning are the approximation hierarchies for the copositive cone developed in the last decade. Here, we only refer to the notable papers [87], [88] and [126]. A common limitation of these uniform approximation hierarchies is that the cost of computation in each hierarchy increases rapidly as the number of hierarchy increases. Hence, they are not suitable for detecting the copositivity of medium or large size matrices.

As Bomze and Eichfelder pointed out in [25], “there are but a few implemented numerical algorithms which apply to general symmetric matrices without any structural assumptions or dimensional restrictions and are not merely recursive but rather focus on generating subproblems in a somehow data-driven way.” In this chapter, we present a new recursive algorithm dealing with both issues as mentioned in the quote. The subproblem size of our algorithm does not increase too fast, as it iterates and an adaptive scheme is adopted such that the information in the matrix data is embedded in each iteration.

This rest of this chapter is arranged as follows. In Section 3.2, we introduce some properties of the cone of nonnegative quadratic functions, which is the basic ingredient of our algorithm. In Section 3.3, the quadratic optimization problem over the standard simplex for copositivity determination is transformed into an equivalent linear conic programming problem, which is then approximated by another solvable linear conic programming problem defined on the dual of the cone of nonnegative quadratic functions over a union of ellipsoids. The linear matrix inequalities (LMI) representation of this cone is also presented. In Section 3.4, an adaptive scheme is designed to refine the union of ellipsoids, and the finite termination of the proposed algorithm is proved. At last, some numerical results are provided to illustrate the validity and efficiency of the proposed algorithm.

3.2 The cone of nonnegative quadratic functions

In Chapter 2, we introduced the cone of nonnegative quadratic functions, $\mathcal{N}_{\mathcal{F}}$, over a given set $\mathcal{F}$. In this section, we further study the properties of this cone, especially for some special cases of $\mathcal{F}$.

For any nonempty set $\mathcal{F} \subseteq \mathbb{R}^n$, it is obvious that $\mathcal{S}_+^{n+1} \subseteq \mathcal{N}_{\mathcal{F}}$ by definition of (2.7). Since $\texttt{int}(\mathcal{S}_+^{n+1}) \neq \emptyset$, $\mathcal{N}_{\mathcal{F}}$ is always solid. The next theorem offers the property of the boundary points of $\mathcal{N}_{\mathcal{F}}$.

Theorem 3.2.1. *Assume the nonempty set $\mathcal{F} \subseteq \mathbb{R}^n$ is closed and bounded. For a given matrix $U \in \mathcal{N}_{\mathcal{F}}$, the following three statements are equivalent:*
(1) U is a boundary point of $\mathcal{N}_{\mathcal{F}}$;
(2) $f_U(x) = \begin{bmatrix} 1 \\ x \end{bmatrix}^T U \begin{bmatrix} 1 \\ x \end{bmatrix} \geq 0$ for any $x \in \mathcal{F}$ and there exists at least one $\bar{x} \in \mathcal{F}$ such that $f_U(\bar{x}) = 0$;
(3) $U \in \mathcal{N}_{\mathcal{F}}$, and $U - \sigma \begin{bmatrix} 1 & \mathbf{0}_{1\times n} \\ \mathbf{0}_{n\times 1} & \mathbf{0}_{n\times n} \end{bmatrix} \notin \mathcal{N}_{\mathcal{F}}$ for any $\sigma > 0$, where $\mathbf{0}_{m\times n}$ is an $m \times n$ matrix of all zeros.

Proof. (1) $\Rightarrow$ (2) Assume U is a boundary point of $\mathcal{N}_{\mathcal{F}}$, then $f_U(x) \geq 0$ for all $x \in \mathcal{F}$ by definition. If $f_U(x) > 0$ for all $x \in \mathcal{F}$, since $\mathcal{F}$ is bounded and closed, then $\min_{x\in\mathcal{F}} f_U(x) = \beta > 0$. Denote $\gamma = \max \left\{ \|Z\| \middle| Z = \begin{bmatrix} 1 & x^T \\ x & xx^T \end{bmatrix}, x \in \mathcal{F} \right\}$, where $\|Z\| = \sqrt{Z \bullet Z}$. Since $\mathcal{F}$ is bounded, then $\gamma < +\infty$. Therefore, for any real symmetric matrix $U_0 \in \mathcal{S}^{n+1}$ such that $\|U_0\| < \frac{\beta}{\gamma}$, we have $f_{U+U_0}(x) = \begin{bmatrix} 1 \\ x \end{bmatrix}^T U \begin{bmatrix} 1 \\ x \end{bmatrix} + \begin{bmatrix} 1 \\ x \end{bmatrix}^T U_0 \begin{bmatrix} 1 \\ x \end{bmatrix} > \beta - \gamma\frac{\beta}{\gamma} = 0$ for all $x \in \mathcal{F}$. Thus, $U + U_0 \in \mathcal{N}_{\mathcal{F}}$ and U is an interior point of $D_{\mathcal{F}}$, which contradicts the assumption. Therefore, there exists at least one $\bar{x} \in \mathcal{F}$ such that $f_U(\bar{x}) = 0$.
(2) $\Rightarrow$ (3) If $f_U(x) \geq 0$ for any $x \in \mathcal{F}$ and $f_U(\bar{x}) = 0$ for some $\bar{x} \in \mathcal{F}$, then $f_U(\bar{x}) - \sigma < 0$ for any $\sigma > 0$, which is equivalent to $U - \sigma \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \notin \mathcal{N}_{\mathcal{F}}$ for any $\sigma > 0$.
(3) $\Rightarrow$ (1) This follows from the definition of boundary points. □

A proper cone is important for numerical stability and feasibility of an algorithm. A natural question is when will the cone $\mathscr{N}_{\mathcal{F}}$ be proper. The answer is given in the next theorem.

Theorem 3.2.2 ([74])**.** *If the set $\mathcal{F} \subseteq \mathbb{R}^n$ has a nonempty interior, then both of the cone $\mathscr{N}_{\mathcal{F}}$ and its dual cone $\mathscr{N}_{\mathcal{F}}^*$ are proper.*

3.2.1 A special case of $\mathcal{F}$

In this subsection, we specifically consider a special case of $\mathcal{F} = \mathcal{E} = \{x \in \mathbb{R}^n | x^T Ax + 2b^T x + c \leq 0\}$, which is a full-dimensional ellipsoid. The next theorem, which is equivalent to Theorem 2.3.1, is useful when we develop the linear matrix inequality (LMI) representations of $\mathscr{N}_{\mathcal{E}}$ and $\mathscr{N}_{\mathcal{E}}^*$.

Theorem 3.2.3. *Given any $A \in \mathcal{S}^n_{++}$, $b \in \mathbb{R}^n$, $c \in \mathbb{R}$ and a nonzero matrix $Y \in \mathcal{S}^{n+1}$ that satisfies*

$$\begin{bmatrix} c & b^T \\ b & A \end{bmatrix} \bullet Y \leq 0, \quad Y \in \mathcal{S}^{n+1}_+,$$

there exists a rank-one decomposition of Y such that

$$Y = \sum_{i=1}^{r} \alpha_i y^i (y^i)^T \tag{3.1}$$

with some $r \in \mathbb{N}$, $\alpha_i > 0$, $y^i = \begin{bmatrix} 1 \\ x^i \end{bmatrix}$ and $(x^i)^T Ax^i + 2b^T x^i + c \leq 0$ for $i = 1, 2, ..., r$.

As we have shown in Theorem 2.3.1, this decomposition can be done in polynomial time ([117] and [125]). Now we can prove the next theorem.

Theorem 3.2.4. *For a full-dimensional ellipsoid $\mathcal{E} = \{x \in \mathbb{R}^n | x^T Ax + 2b^T x + c \leq 0\}$ where $A \in \mathcal{S}^n_{++}$, $b \in \mathbb{R}^n$ and $c \in \mathbb{R}$, we have*

$$\begin{aligned} \mathscr{N}_{\mathcal{E}}^* &= \texttt{cone}\left\{ yy^T \in \mathcal{S}^{n+1} \middle| y = \begin{bmatrix} 1 \\ x \end{bmatrix}, x \in \mathcal{E} \right\} \\ &= \left\{ Y \in \mathcal{S}^{n+1}_+ \middle| \begin{bmatrix} c & b^T \\ b & A \end{bmatrix} \bullet Y \leq 0 \right\}. \end{aligned} \tag{3.2}$$

Proof. The first equation holds due to (2.11) and the fact that $\mathcal{E}$ is closed and bounded. We only need to prove the second equation. Suppose that $Y = \sum_{i=1}^r \alpha_i y^i (y^i)^T \in \mathcal{S}^{n+1}_+$ ($Y \neq 0$) with $y^i = \begin{bmatrix} 1 \\ x^i \end{bmatrix}$, $x^i \in \mathcal{E}$ and $\alpha_i > 0$ for $i = 1, ..., r$ ($r \in \mathbb{N}$). Then, $(y^i)^T \begin{bmatrix} c & b^T \\ b & A \end{bmatrix} y^i \leq 0$ for all

y^i. Consequently,

$$\begin{bmatrix} c & b^T \\ b & A \end{bmatrix} \bullet Y = \sum_{i=1}^{r} \alpha_i (y^i)^T \begin{bmatrix} c & b^T \\ b & A \end{bmatrix} y^i \leq 0.$$

Conversely, for any $Y \in \mathcal{S}_+^{n+1}$ ($Y \neq 0$) with $\begin{bmatrix} c & b^T \\ b & A \end{bmatrix} \bullet Y \leq 0$, there exists $y^i = \begin{bmatrix} 1 \\ x^i \end{bmatrix}$, where $(x^i)^T A x^i + 2b^T x^i + c \leq 0$ and $\alpha_i > 0$ for $i = 1, ..., r$ ($r \in \mathbb{N}$), such that $Y = \sum_{i=1}^r \alpha_i y^i (y^i)^T$ according to Theorem 3.2.3. Therefore, $x^i \in \mathcal{E}$. This competes the proof. □

Theorem 3.2.4 gives the LMI representation of $\mathscr{N}_{\mathcal{E}}^*$ and the following theorem gives the LMI representation of $\mathscr{N}_{\mathcal{E}}$. The proof is the same as Corollary 5 of [117].

Theorem 3.2.5. *For a full-dimensional ellipsoid* $\mathcal{E} = \{x \in \mathbb{R}^n | x^T A x + 2b^T x + c \leq 0\}$ *where* $A \in \mathcal{S}_{++}^n$, $b \in \mathbb{R}^n$ *and* $c \in \mathbb{R}$, *we have*

$$\mathscr{N}_{\mathcal{E}} = \left\{ U \in \mathcal{S}^{n+1} \middle| U + \lambda \begin{bmatrix} c & b^T \\ b & A \end{bmatrix} \in \mathcal{S}_+^{n+1} \text{ for some } \lambda \geq 0 \right\}. \tag{3.3}$$

3.3 Conic Reformulation and Approximation Cones

To detect whether a given matrix is copositive, we can formulate this problem as an equivalent quadratic programming problem over the standard simplex $\mathcal{F}_\Delta$ and then reformulate the problem as a linear conic programming problem over the cone $\mathscr{N}_{\mathcal{F}_\Delta}^*$. Since there is no known efficient algorithm to check whether a matrix is in the cone $\mathscr{N}_{\mathcal{F}_\Delta}^*$, the cone $\mathscr{N}_{\mathcal{F}_\Delta}^*$ is *uncomputable*. Thus, we introduce a new cone $\mathscr{N}_{\mathscr{E}}^*$, the dual of the cone of nonnegative quadratic functions over a union of ellipsoids $\mathscr{E}$, to approximate the cone $\mathscr{N}_{\mathcal{F}_\Delta}^*$ and present some important properties of $\mathscr{N}_{\mathscr{E}}$ and $\mathscr{N}_{\mathscr{E}}^*$.

3.3.1 Conic reformulation

Recall that a matrix $M \in \mathcal{S}^n$ is copositive if its homogeneous quadratic form $f(x) = x^T M x \geq 0$ for all $x \in \mathbb{R}_+^n$. Therefore, M is copositive if and only if the optimal value $V(\text{StQP}) \geq 0$ for the following problem:

$$\text{(StQP)} \qquad \begin{array}{ll} \min & x^T M x \\ \text{s.t.} & x \in \mathcal{F}_\Delta = \left\{ x \in \mathbb{R}^n | e^T x = 1, x \geq 0 \right\}, \end{array} \tag{3.4}$$

where $e = (1, \cdots, 1)^T \in \mathbb{R}^n$. The problem is the so-called standard quadratic programming problem. Some references for solving the problem are [23], [24] and [70]. However, our aim here is to determine the sign of the optimal value in order to detect the copositivity of matrix M, not the exact optimal value. Therefore, we do not need to solve problem (StQP) exactly, but to obtain a good estimation. In the rest of this subsection, we will reformulate problem (StQP) to an equivalent linear conic programming problem, as shown in Section 2.2. Define the cone of nonnegative quadratic functions over $\mathcal{F}_\Delta$ as

$$\mathscr{N}_{\mathcal{F}_\Delta} = \left\{ U \in \mathcal{S}^{n+1} \middle| \begin{bmatrix} 1 \\ x \end{bmatrix}^T U \begin{bmatrix} 1 \\ x \end{bmatrix} \geq 0 \text{ for all } x \in \mathcal{F}_\Delta \right\}, \tag{3.5}$$

and define the set

$$\mathcal{Z}_{\mathcal{F}_\Delta} = \left\{ Y \in \mathcal{S}^{n+1} \middle| Y = \begin{bmatrix} 1 \\ x \end{bmatrix} \begin{bmatrix} 1 \\ x \end{bmatrix}^T \text{ for some } x \in \mathcal{F}_\Delta \right\}. \tag{3.6}$$

Moreover, let $\texttt{cone}\{\mathcal{Z}_{\mathcal{F}_\Delta}\} = \{Y \in \mathcal{S}^{n+1} | Y = \alpha_1 Z_1 + \cdots + \alpha_r Z_r \text{ for some } r \in \mathbb{N},\ \alpha_i \geq 0,\ Z_i \in \mathcal{Z}_{\mathcal{F}_\Delta}, i = 1, ..., r\}$ be the conic hull of the set $\mathcal{Z}_{\mathcal{F}_\Delta}$. Since $\mathcal{F}_\Delta$ is closed and bounded,

$$\mathscr{N}^*_{\mathcal{F}_\Delta} = \texttt{cone}\{\mathcal{Z}_{\mathcal{F}_\Delta}\} \tag{3.7}$$

is the dual cone of $\mathscr{N}_{\mathcal{F}_\Delta}$ according to (2.11). From Section 2.2, problem (StQP) is equivalent to the following linear conic programming problem (CP-StQP):

$$\text{(CP-StQP)} \qquad \begin{aligned} \min \quad & H \bullet Y \\ \text{s.t.} \quad & Y_{11} = 1, \\ & Y \in \mathscr{N}^*_{\mathcal{F}_\Delta}, \end{aligned} \tag{3.8}$$

where matrix $H = \begin{bmatrix} 0 & 0 \\ 0 & M \end{bmatrix} \in \mathcal{S}^{n+1}$ and Y_{11} is the first entry of matrix Y. By the linear conic duality theory in [20], the dual of the problem (CP-StQP) is defined by

$$\text{(CD-StQP)} \qquad \begin{aligned} \max \quad & \sigma \\ \text{s.t.} \quad & \begin{bmatrix} -\sigma & 0 \\ 0 & M \end{bmatrix} \in \mathscr{N}_{\mathcal{F}_\Delta},\ \sigma \in \mathbb{R}. \end{aligned} \tag{3.9}$$

Since the set $\mathcal{F}_\Delta$ is nonempty, problem (CP-StQP) is always feasible. Also, for any given matrix M, we can choose $\bar{\sigma}$ small enough such that $\bar{\sigma} < V(\text{StQP}) < +\infty$ due to the fact that the set $\mathcal{F}_\Delta$ is closed and bounded. Then $\begin{bmatrix} -\bar{\sigma} & 0 \\ 0 & M \end{bmatrix}$ is an interior point of $\mathscr{N}_{\mathcal{F}_\Delta}$ according to Theorem 3.2.1 because $\begin{bmatrix} 1 \\ x \end{bmatrix}^T \begin{bmatrix} -\bar{\sigma} & 0 \\ 0 & M \end{bmatrix} \begin{bmatrix} 1 \\ x \end{bmatrix} = x^T M x - \bar{\sigma} > 0$ for all $x \in \mathcal{F}_\Delta$. Thus, the problem (CD-

StQP) is always strictly feasible. Then, the duality gap between the problems (CP-StQP) and (CD-StQP) is zero by Theorem 2.4.2. Consequently, we have the next theorem.

Theorem 3.3.1. *Problem* (CP-StQP) *is feasible, problem* (CD-StQP) *is strictly feasible, the optimal values of problems* (StQP), (CP-StQP) *and* (CD-StQP) *are equal, and the optimal solution of problem* (CP-StQP) *is attainable.*

The attainability of problem (CP-StQP) is due to the strong feasibility of problem (CD-StQP).

According to Theorem 3.3.1, both of the problems (CP-StQP) and (CD-StQP) are equivalent to problem (StQP), thus are NP-hard. In fact, we do not know any efficient algorithm to determine whether a given matrix is in $\mathscr{N}^*_{\mathcal{F}_\Delta}$ or not. Besides, $\mathscr{N}^*_{\mathcal{F}_\Delta}$ is not solid because $\mathscr{N}_{\mathcal{F}_\Delta}$ is not pointed due to the fact that $\begin{bmatrix} 2 & -e^T \\ -e & 0 \end{bmatrix} \in \mathscr{N}_{\mathcal{F}_\Delta}$ and $-\begin{bmatrix} 2 & -e^T \\ -e & 0 \end{bmatrix} \in \mathscr{N}_{\mathcal{F}_\Delta}$. Hence, the optimal solution of the problem (CD-StQP) may not be attainable. Both of these disadvantages suggest to solve this problem via approximating $\mathscr{N}_{\mathcal{F}_\Delta}$ and $\mathscr{N}^*_{\mathcal{F}_\Delta}$ by other proper cones, such as the positive semidefinite cone $\mathcal{S}^{n+1}_+$. However, the approximations by using $\mathcal{S}^{n+1}_+$ may not be tight enough [93]. If we can design an alternative $\mathscr{N}_{\mathscr{E}}$, such that $\mathcal{S}^{n+1}_+ \subseteq \mathscr{N}_{\mathscr{E}} \subseteq \mathscr{N}_{\mathcal{F}_\Delta}$ and $\mathcal{S}^{n+1}_+ \supseteq \mathscr{N}^*_{\mathscr{E}} \supseteq \mathscr{N}^*_{\mathcal{F}_\Delta}$, then a tighter lower bound of problem (StQP) may be obtained via replacing the cone $\mathscr{N}^*_{\mathcal{F}_\Delta}$ in problem (CP-StQP) by the cone $\mathscr{N}^*_{\mathscr{E}}$ instead of $\mathcal{S}^{n+1}_+$. In the next subsection, we will describe such cones and their related properties. Moreover, it is the sign of the optimal value V(StQP) that matters in deciding whether the given matrix M is copositive or not. Therefore, if the lower bound is good enough to determine the sign of V(StQP), it would be more efficient to obtain this bound than to solve problem (StQP) exactly.

3.3.2 LMI based approximation cones

Let

$$\mathbf{E} = \bigcup_{i=1}^{K} \{\mathcal{E}_i\} \tag{3.10}$$

be a collection of full-dimensional ellipsoids $\mathcal{E}_i$, where each

$$\mathcal{E}_i = \{x \in \mathbb{R}^n | x^T A^i x + 2(b^i)^T x + c^i \leq 0\} \tag{3.11}$$

with $A^i \in \mathcal{S}^n_{++}$, $b^i \in \mathbb{R}^n$ and $c^i \in \mathbb{R}$ for $i = 1, 2, ..., k$. Let $\mathscr{E}$ be the union of the ellipsoids in $\mathbf{E}$. We say $\mathscr{E}$ is an *elliptic cover* of $\mathcal{F}_\Delta$ if

$$\mathcal{F}_\Delta \subseteq \mathscr{E} = \bigcup_{i=1}^{K} \mathcal{E}_i. \tag{3.12}$$

From Subsection 3.2.1, each cone $\mathscr{N}_{\mathcal{E}_i}$ has an LMI representation. Define the cone of nonnegative quadratic functions over the set $\mathscr{E}$ as

$$\mathscr{N}_{\mathscr{E}} = \left\{ U \in \mathcal{S}^{n+1} \middle| \begin{bmatrix} 1 \\ x \end{bmatrix}^T U \begin{bmatrix} 1 \\ x \end{bmatrix} \geq 0 \text{ for all } x \in \mathscr{E} \right\}, \tag{3.13}$$

and its dual cone

$$\mathscr{N}_{\mathscr{E}}^* = \texttt{cone} \left\{ Y \in \mathcal{S}^{n+1} \middle| Y = \begin{bmatrix} 1 \\ x \end{bmatrix} \begin{bmatrix} 1 \\ x \end{bmatrix}^T \text{ for some } x \in \mathscr{E} \right\}. \tag{3.14}$$

Since each $\mathcal{E}_i$ has a nonempty interior, so does $\mathscr{E}$. Thus, $\mathscr{N}_{\mathscr{E}}$ and $\mathscr{N}_{\mathscr{E}}^*$ are proper according to Theorem 3.2.2. Besides, according to the properties of the dual cone stated in Section 2.2, it is easy to show the following properties:

Theorem 3.3.2. *If $\mathcal{F}_\Delta \subseteq \mathscr{E}$, then $\mathscr{N}_{\mathcal{F}_\Delta} \supseteq \mathscr{N}_{\mathscr{E}}$ and $\mathscr{N}_{\mathcal{F}_\Delta}^* \subseteq \mathscr{N}_{\mathscr{E}}^*$.*

Theorem 3.3.3. *If $\mathscr{E} = \mathcal{E}_1 \cup \mathcal{E}_2 \cup \cdots \cup \mathcal{E}_k$, then $\mathscr{N}_{\mathscr{E}} = \mathscr{N}_{\mathcal{E}_1} \cap \mathscr{N}_{\mathcal{E}_2} \cap \cdots \cap \mathscr{N}_{\mathcal{E}_k}$ and $\mathscr{N}_{\mathscr{E}}^* = \mathscr{N}_{\mathcal{E}_1}^* + \mathscr{N}_{\mathcal{E}_2}^* + \cdots + \mathscr{N}_{\mathcal{E}_k}^* = \{x^1 + x^2 + \cdots + x^k | x^1 \in \mathcal{E}_1,\ x^2 \in \mathcal{E}_2, \ldots, x^k \in \mathcal{E}_k\}$.*

From Theorem 3.3.3, we have some LMI representations of $\mathscr{N}_{\mathscr{E}}$ and $\mathscr{N}_{\mathscr{E}}^*$.

Corollary 3.3.4. *Let sets $\mathcal{E}_i$, $i = 1, ..., k$, $\mathscr{E}$, $\mathscr{N}_{\mathscr{E}}$ and $\mathscr{N}_{\mathscr{E}}^*$ be defined by* (3.12)-(3.14). *Then for any $X \in \mathcal{S}^{n+1}$, we have $X \in \mathscr{N}_{\mathscr{E}}$ if and only if*

$$X + \lambda_i \begin{bmatrix} c^i & (b^i)^T \\ b^i & A^i \end{bmatrix} \in \mathcal{S}_+^{n+1} \text{ for some } \lambda_i \geq 0 \tag{3.15}$$

holds for all $i = 1, ..., k$. And, for any $Y \in \mathcal{S}^{n+1}$, we have $Y \in \mathscr{N}_{\mathscr{E}}^*$ if and only if

$$\begin{aligned} &Y = Y^1 + Y^2 + \cdots + Y^k, \\ &\begin{bmatrix} c^i & (b^i)^T \\ b^i & A^i \end{bmatrix} \bullet Y^i \leq 0,\ Y^i \in \mathcal{S}_+^{n+1} \text{ for } i = 1, 2, \ldots, k. \end{aligned} \tag{3.16}$$

Proof. This result is a direct consequence of Theorems 3.2.4, 3.2.5 and 3.3.3. □

Based on Theorem 3.3.2, if we can design a union of ellipsoids $\mathscr{E} \supseteq \mathcal{F}_\Delta$ such that $\mathscr{E}$ is close to $\mathcal{F}_\Delta$, then $\mathscr{N}_{\mathscr{E}}^*$ is a good approximation of $\mathscr{N}_{\mathcal{F}_\Delta}^*$. In this case, we can use $\mathscr{N}_{\mathscr{E}}^*$ to replace $\mathscr{N}_{\mathcal{F}_\Delta}^*$ in the problem (CP-StQP) and obtain a good lower bound. In the next section, we will show how to generate and refine such $\mathscr{E}$ in details.

3.4 Conic Approximation to Problem (StQP)

In this section, we study how to use the cones proposed in Subsection 3.3.2 to approximate problem (CP-StQP).

Assume $\mathscr{E} = \mathcal{E}_1 \cup \cdots \cup \mathcal{E}_k$ is an elliptic cover of $\mathcal{F}_\Delta$ as defined in (3.12). Then, $\mathscr{N}_{\mathscr{E}}^*$ can be used to approximate $\mathscr{N}_{\mathcal{F}_\Delta}^*$. Before relaxing the problem (CP-StQP), we rewrite it in the following form:

$$\text{(CP-StQP)} \qquad \begin{array}{ll} \min & H \bullet Y \\ \text{s.t.} & \begin{bmatrix} 2 & -e^T \\ -e & 0 \end{bmatrix} \bullet Y = 0, \\ & Y_{11} = 1, \\ & Y \in \mathscr{N}_{\mathcal{F}_\Delta}^* \end{array} \tag{3.17}$$

by adding a redundant constraint

$$\begin{bmatrix} 2 & -e^T \\ -e & 0 \end{bmatrix} \bullet Y = 0.$$

To verify the previous constraint is redundant, notice that, according to the definition of $\mathscr{N}_{\mathcal{F}_\Delta}^* = \texttt{cone}\{Z_{\mathcal{F}_\Delta}\}$, any nonzero matrix $Y \in \mathscr{N}_{\mathcal{F}_\Delta}^*$ can be decomposed into $Y = \sum_{i=1}^r \alpha_i y^i (y^i)^T$, where $y^i = \begin{bmatrix} 1 \\ x^i \end{bmatrix}$ with $x^i \in \mathcal{F}_\Delta$ for some $r \in \mathbb{N}$, $\alpha_i > 0$ and $\begin{bmatrix} 2 & -e^T \\ -e & 0 \end{bmatrix} \bullet (y^i (y^i)^T) = 0$ for $i = 1, 2, ..., r$. Notice that this redundant constraint may not be redundant in the following relaxed conic programming problem (RCP-StQP):

$$\text{(RCP-StQP)} \qquad \begin{array}{ll} \min & H \bullet Y \\ \text{s.t.} & \begin{bmatrix} 2 & -e^T \\ -e & 0 \end{bmatrix} \bullet Y = 0, \\ & Y_{11} = 1, \\ & Y \in \mathscr{N}_{\mathscr{E}}^*. \end{array} \tag{3.18}$$

The purpose of adding the extra constraint in problem (RCP-StQP) is to improve the lower bound of problem (CP-StQP). The dual problem of (RCP-StQP) is defined as

$$\text{(RCD-StQP)} \qquad \begin{array}{ll} \max & \sigma \\ \text{s.t.} & \begin{bmatrix} -\sigma & 0 \\ 0 & M \end{bmatrix} - \mu \begin{bmatrix} 2 & -e^T \\ -e & 0 \end{bmatrix} \in \mathscr{N}_{\mathscr{E}}, \\ & \mu \in \mathbb{R},\ \sigma \in \mathbb{R} \end{array} \tag{3.19}$$

Using Corollary 3.3.4, we can rewrite problem (RCP-StQP) and problem (RCD-StQP) in the following specific forms:

$$
\text{(RCP-StQP)} \qquad
\begin{array}{ll}
\min & H \bullet Y \\
\text{s.t.} & \begin{bmatrix} 2 & -e^T \\ -e & 0 \end{bmatrix} \bullet Y = 0 \\
& Y_{11} = 1 \\
& Y = Y^1 + Y^2 + \cdots + Y^k \\
& \begin{bmatrix} c^i & (b^i)^T \\ b^i & A^i \end{bmatrix} \bullet Y^i \leq 0, \; Y^i \in \mathcal{S}_+^{n+1} \text{ for } i = 1, 2, \ldots, k
\end{array}
\tag{3.20}
$$

and

$$
\text{(RCD-StQP)} \qquad
\begin{array}{ll}
\max & \sigma \\
\text{s.t.} & S = \begin{bmatrix} -\sigma & 0 \\ 0 & M \end{bmatrix} - \mu \begin{bmatrix} 2 & -e^T \\ -e & 0 \end{bmatrix} \\
& S + \lambda_i \begin{bmatrix} c^i & (b^i)^T \\ b^i & A^i \end{bmatrix} \in \mathcal{S}_+^{n+1} \text{ for } i = 1, 2, ..., k, \\
& \mu \in \mathbb{R}, \; \sigma \in \mathbb{R}, \; \lambda_i \geq 0 \text{ for } i = 1, 2, ..., k.
\end{array}
\tag{3.21}
$$

Although problem (CP-StQP) is not strictly feasible because the cone $\mathcal{N}^*_{\mathcal{F}_\Delta}$ is not solid, problems (RCP-StQP) and (RCD-StQP) are both strictly feasible under some mild condition.

Theorem 3.4.1. *If each ellipsoid $\mathcal{E}_i = \{x \in \mathbb{R}^n | x^T A^i x + 2(b^i)^T x + c^i \leq 0\}$ in $\mathbf{E}$ has an interior point falling on the hyperplane $\Pi = \{x \in \mathbb{R}^n | e^T x = 1\}$, i.e., there is a point $\bar{x}^i \in \mathbb{R}^n$ such that $e^T \bar{x}^i = 1$ and $(\bar{x}^i)^T A^i \bar{x}^i + 2(b^i)^T \bar{x}^i + c^i < 0$, then problem* (RCP-StQP) *is strictly feasible. Moreover, problem* (RCD-StQP) *is always strictly feasible, the optimal solutions of problems* (RCP-StQP) *and* (RCD-StQP) *are attainable and there is no duality gap between problems* (RCP-StQP) *and* (RCD-StQP).

Proof. Because $\bar{x}^i$ is an interior point of $\mathcal{E}_i$, there exists an n-dimensional simplex with affinely independent vertices $\bar{x}^{ij}$, $j = 1, \ldots, n+1$, contained in the interior of the ellipsoid $\mathcal{F}_i$ such that $\bar{x}^i$ is an interior point of this simplex. Then $\bar{x}^i = \sum_{j=1}^{n+1} \bar{\alpha}_{ij} \bar{x}^{ij}$ with $\bar{\alpha}_{ij} > 0$ for $j = 1, \ldots, n+1$ and $\sum_{j=1}^{n+1} \bar{\alpha}_{ij} = 1$. Consider the matrix

$$
\bar{Y}^i = \frac{1}{k} \sum_{j=1}^{n+1} \bar{\alpha}_{ij} \begin{bmatrix} 1 \\ \bar{x}^{ij} \end{bmatrix} \begin{bmatrix} 1 \\ \bar{x}^{ij} \end{bmatrix}^T
$$

for $i = 1, \ldots, k$. It is easy to check that $\begin{bmatrix} 2 & -e^T \\ -e & 0 \end{bmatrix} \bullet \bar{Y}^i = 0$. Also, $\bar{Y}^i \in \mathcal{S}_{++}^{n+1}$ because the vertices $\bar{x}^{ij}$, $j = 1, ..., n+1$, are affinely independent. Let $\bar{Y} = \sum_{i=1}^{k} \bar{Y}^i$, then $(\bar{Y}^1, \ldots, \bar{Y}^k, \bar{Y})$ is a strictly feasible solution to problem (RCP-StQP). This proves the strong feasibility of problem (RCP-StQP). For any matrix M and $\bar{\mu}$ in problem (RCD-StQP), note that $A^i \in \mathcal{S}_{++}^{n+1}$, for $i = 1, \ldots, k$, there exists a large enough $\bar{\lambda}_i > 0$ and a small enough $\bar{\sigma}$ such that the matrix

$$\bar{S} = \begin{bmatrix} -\bar{\sigma} & 0 \\ 0 & M \end{bmatrix} - \bar{\mu} \begin{bmatrix} 2 & -e^T \\ -e & 0 \end{bmatrix}$$

satisfies that

$$\bar{S} + \bar{\lambda}_i \begin{bmatrix} c^i & (b^i)^T \\ b^i & A^i \end{bmatrix} \in \mathcal{S}_{++}^{n+1} \text{ for } i = 1, \ldots, k,$$

due to the diagonal dominance. This proves the strong feasibility of problem (RCD-StQP). The rest of the claims hold according to Theorem 2.4.2. □

By the linear conic optimality conditions in Corollary 2.4.3, we have the following optimality condition for problems (RCP-StQP) and (RCD-StQP):

$$\begin{aligned} &(Y^1, Y^2, \ldots, Y^k, Y) \text{ is feasible to problem (RCP-StQP)}, \\ &(\sigma, \mu, \lambda, S) \text{ is feasible to problem (RCD-StQP)}, \qquad \text{(Optimality Conditions)} \\ &S \bullet Y^i = 0, \;\; \lambda_i \begin{bmatrix} c^i & (b^i)^T \\ b^i & A^i \end{bmatrix} \bullet Y^i = 0 \text{ for } i = 1, 2, \ldots, k. \end{aligned}$$

The number of free variables in the problem (RCP-StQP) (including $Y^1, ..., Y^k$) is mkn^2 for some $m \in \mathbb{N}$. However, it is inefficient to introduce too many ellipsoids to approximate the set $\mathcal{F}_\Delta$ well enough everywhere. Therefore, we need to design an efficient arrangement of ellipsoids $\mathcal{E}_i$'s to cover $\mathcal{F}_\Delta$. In the next sections, an adaptive scheme is introduced to achieve this purpose.

3.5 An Adaptive Scheme for Detecting Copositive Matrices

3.5.1 Sensitive points and sensitive ellipsoids

In this subsection, the definitions of a sensitive point and a sensitive ellipsoid are given to indicate which ellipsoid $\mathcal{E}_i$ in $\mathbf{E}$ should be refined. In order to detect such an ellipsoid, the next result is needed.

Corollary 3.5.1. *If* $Y^* = (Y^1)^* + \cdots + (Y^k)^*$ *is an optimal solution of problem* (RCP-StQP)*,* *then, for $(Y^i)^* \neq 0$ with $i \in \{1, 2, \dots, k\}$, we have*

$$(Y^i)^* = \sum_{j=1}^{n_i} \alpha_{ij} \begin{bmatrix} 1 \\ x^{ij} \end{bmatrix} \begin{bmatrix} 1 \\ x^{ij} \end{bmatrix}^T \tag{3.22}$$

for some $n_i \in \{1, 2, ..., n+1\}$, $\alpha_{ij} > 0$, $x^{ij} \in \mathcal{E}_i$. Moreover, Y^* can be decomposed into

$$Y^* = \sum_{i:(Y^i)^* \neq 0} \sum_{j=1}^{n_i} \alpha_{ij} \begin{bmatrix} 1 \\ x^{ij} \end{bmatrix} \begin{bmatrix} 1 \\ x^{ij} \end{bmatrix}^T, \tag{3.23}$$

with $\sum_{i:(Y^i)^* \neq 0} \sum_{j=1}^{n_i} \alpha_{ij} = 1$.

Proof. The result is a direct consequence of Theorem 3.2.4. □

All the points x^{ij} in (3.23) are defined as *sensitive points* for problem (RCD-StQP). The optimal value of problem (RCD-StQP) is sensitive to x^{ij} since $\begin{bmatrix} 1 \\ x^{ij} \end{bmatrix}^T \left(S^* + \lambda_i^* \begin{bmatrix} c^i & (b^i)^T \\ b^i & A^i \end{bmatrix} \right) \begin{bmatrix} 1 \\ x^{ij} \end{bmatrix} = 0$ is an active constraint for the optimal solution (S^*, λ^*) meeting the optimality conditions. Also, we define the ellipsoid $\mathcal{E}_i$ containing all these x^{ij}, $j = 1, 2, \dots, n_i$, as a *sensitive ellipsoid*.

Among all sensitive points, we define the *most sensitive point* as follows.

Definition 1. For the rank-one decomposition

$$Y^* = \sum_{i:(Y^i)^* \neq 0} \sum_{j=1}^{n_i} \alpha_{ij} \begin{bmatrix} 1 \\ x^{ij} \end{bmatrix} \begin{bmatrix} 1 \\ x^{ij} \end{bmatrix}^T,$$

x^* is the most sensitive point if

$$x^* = \text{argmin} \left\{ (x^{ij})^T M x^{ij} : x^{ij} \neq 0 \text{ for } i = 1, ..., k \text{ and } j = 1, ..., n_i . \right\}$$

That is, x^* has the minimal objective value among all sensitive points. Note that the most sensitive point x^* may not be unique. If there are multiple most sensitive points, we choose the one with the smallest index in i with smallest j as a tie-breaker, and denote the smallest index i by t. Then, $\mathcal{E}_t$, the ellipsoid which x^* is decomposed from, is called the *most sensitive ellipsoid.*

Theorem 3.5.2. *If Y^* is the optimal solution of problem* (RCP-StQP) *with the most sensitive point x^*, then*

$$u^* = (x^*)^T M x^* \leq V(\text{CP-StQP}) \tag{3.24}$$

If $u^* \geq 0$, then matrix M is copositive. If $x^* \in \mathbb{R}^n_+$ and $u^* < 0$, then matrix M is not copositive. Moreover, if $x^* \in \mathcal{F}_\Delta$, then the matrix $\begin{bmatrix} 1 \\ x^* \end{bmatrix} \begin{bmatrix} 1 \\ x^* \end{bmatrix}^T$ is an optimal solution of problem (CP-StQP) *and x^* is an optimal solution of problem* (StQP).

Proof. The inequality in (3.24) holds because

$$\begin{array}{rcl} V(\text{CP-StQP}) & \geq & V(\text{RCP-StQP}) \\ & = & \sum_{i:(Y^i)^* \neq 0} \sum_{j=1}^{n_i} \alpha_{ij} (x^{ij})^T M x^{ij} \\ & \geq & \sum_{i:(Y^i)^* \neq 0} \sum_{j=1}^{n_i} \alpha_{ij} (x^*)^T M x^* \\ & = & (x^*)^T M x^* \end{array}$$

According to Theorem 3.3.1, if $u^* \geq 0$, then $V(\text{StQP}) = V(\text{CP-StQP}) \geq u^* \geq 0$ and this leads to the claim of the copositivity of matrix M. If $x^* \in \mathbb{R}^n_+ \setminus \{0\}$ and $u^* < 0$, notice that $\bar{x} = x^*/\|x\|_1 \in \mathcal{F}_\Delta$ and $\bar{x}^T M \bar{x} < 0$, this leads to the claim of noncopositivity. If $x^* \in \mathcal{F}_\Delta$, then $\begin{bmatrix} 1 \\ x^* \end{bmatrix} \begin{bmatrix} 1 \\ x^* \end{bmatrix}^T$ is feasible for problem (CP-StQP), and

$$\begin{bmatrix} 1 \\ x^* \end{bmatrix} \begin{bmatrix} 1 \\ x^* \end{bmatrix}^T \bullet H = (x^*)^T M x^* \leq V(\text{CP-StQP}).$$

Hence, matrix $\begin{bmatrix} 1 \\ x^* \end{bmatrix} \begin{bmatrix} 1 \\ x^* \end{bmatrix}^T$ is an optimal solution of problem (CP-StQP) and x^* is an optimal solution of problem (StQP). □

Theorem 3.5.2 shows that if $u^* \geq 0$ or $x^* \in \mathbb{R}^n_+$, then the conclusion about copositivity of matrix M is direct. However, if $u^* < 0$ and $x^* \notin \mathbb{R}^n_+$, then we only obtain a lower bound of problem (StQP) and no conclusion can be drawn. In this case, the current approximate cone $\mathscr{N}^*_{\mathscr{E}}$ is not close enough to $\mathscr{N}^*_{\mathcal{F}_\Delta}$ and the set $\mathscr{E}$ needs to be refined such that the lower bound obtained from problem (RCP-StQP) may be improved.

3.5.2 An adaptive scheme

As mentioned before, fewer ellipsoids involved in the problem (RCP-StQP) is preferred. Therefore, it is unwise to refine $\mathscr{E}$ everywhere. Instead, only the most sensitive ellipsoid $\mathcal{E}_t$ in $\mathscr{E}$ is refined because the most sensitive point x^* in this ellipsoid has the lowest objective value. The basic idea behind the adaptive approximation strategy is that when the most sensitive point x^* and most sensitive ellipsoid $\mathcal{E}_t$ are detected, two ellipsoids constituting a finer cover around x^* replace $\mathcal{E}_t$ in the current set $\mathscr{E}$. By refining the ellipsoids in the region of x^*, we expect to improve the lower bounds of (StQP) significantly. In order to construct and manage the ellipsoids easily, we introduce the following definition.

Definition 2. For a given rectangular set $\mathcal{T} = [u,\ v] = \{x \in \mathbb{R}^n | u_i \leq x_i \leq v_i\}$, define the corresponding *ellipsoid generated by* $\mathcal{T}$ as

$$\mathcal{E}_{\mathcal{T}} = \left\{ x \in \mathbb{R}^n \middle| \sum_{i=1}^{n} \frac{(2x_i - v_i - u_i)^2}{(v_i - u_i)^2} - n \leq 0 \right\}. \tag{3.25}$$

It is easy to see that $\mathcal{E}_{\mathcal{T}}$ is full-dimensional if $u < v$ and $\mathcal{T} \subseteq \mathcal{E}_{\mathcal{T}}$. Similar to the elliptic cover of $\mathcal{F}_{\Delta}$, let

$$\mathbf{T} = \{\mathcal{T}_1\} \cup \cdots \cup \{\mathcal{T}_k\} \tag{3.26}$$

be a collection of full-dimensional rectangular sets $\mathcal{T}_i = [u^i,\ v^i]$ with $u^i < v^i$ for $i = 1, \ldots, k$. Then we define $\mathscr{T}$, the *rectangular set cover* of $\mathcal{F}_{\Delta}$, to be the union of the rectangular sets covering the set $\mathcal{F}_{\Delta}$, that is

$$\mathcal{F}_{\Delta} \subseteq \mathscr{T} = \mathcal{T}_1 \cup \ldots \cup \mathcal{T}_k. \tag{3.27}$$

Other concepts to be used in proving the finite termination of our algorithm are given as follows.

Definition 3. For any set $\mathcal{H} \subseteq \mathbb{R}^n$ and $\delta > 0$, the δ-neighborhood of $\mathcal{H}$ is defined as $\mathcal{B}_{\delta,\mathcal{H}} = \{x \in \mathbb{R}^n | \exists y \in \mathcal{H},\ s.t.\ \|x - y\|_\infty \leq \delta\}$, where $\|\cdot\|_\infty$ means the infinity norm.

Definition 4 ([31])**.** A matrix $M \in \mathcal{S}^n$ is ϵ-copositive if $V(\text{StQP}) \geq -\epsilon$ for some given $\epsilon > 0$.

Definition 5 ([25])**.** A vector $x \in \mathbb{R}^n_+$ is a violating vector for matrix M if $x^T M x < 0$.

First, we need to find an initial ellipsoid $\mathcal{E}_1$ that covers the standard simplex $\mathcal{F}_{\Delta}$. We can set $\mathcal{E}_1 = \{x \in \mathbb{R}^n | \sum_{i=1}^n (2x_i - 1)^2 \leq n\}$, which is generated by the rectangular set $\mathcal{T}_1 = [u^1,\ v^1]$ with $u^1_i = 0$, $v^1_i = 1$ for $i = 1, \ldots, n$. Let the initial rectangular set cover of $\mathcal{F}_{\Delta}$ be $\mathscr{T} = \mathcal{T}_1$ and the initial elliptic cover of $\mathcal{F}_{\Delta}$ be $\mathscr{E} = \mathcal{E}_1$.

When the most sensitive point x^* and the most sensitive ellipsoid $\mathcal{E}_t$ are detected, the rectangular set generating the most sensitive ellipsoid $\mathcal{T}_t = [u^t,\ v^t]$ is also detected. Then, this rectangular set is divided along the direction indicated by the most negative component of x^*. Specifically, denote $id = \min\{\text{argmin}_{\{i=1,\ldots,n\}}\{x^*_i\}\}$, then $\mathcal{T}_t$ is divided into $\mathcal{T}_{t_1} = [u^{t_1},\ v^{t_1}]$ and $\mathcal{T}_{t_2} = [u^{t_2},\ v^{t_2}]$, where $u^{t_1} = u^t$, $v^{t_2} = v^t$, $v^{t_1}_i = v^t_i$, $u^{t_2}_i = u^t_i$ for $i \neq id$ and $v^{t_1}_{id} = u^{t_2}_{id} = (u^t_{id} + v^t_{id})/2$. And the two ellipsoids $\mathcal{E}_{t_1}$ and $\mathcal{E}_{t_2}$ are generated from $\mathcal{T}_{t_1}$ and $\mathcal{T}_{t_2}$ by

$$\mathcal{E}_{t_1} = \left\{ x \in \mathbb{R}^n \middle| \sum_{i=1}^{n} \frac{(2x_i - v^{t_1}_i - u^{t_1}_i)^2}{(v^{t_1}_i - u^{t_1}_i)^2} \leq n \right\}, \tag{3.28}$$

and

$$\mathcal{E}_{t_2} \quad = \quad \left\{ x \in \mathbb{R}^n \middle| \sum_{i=1}^{n} \frac{(2x_i - v_i^{t_2} - u_i^{t_2})^2}{(v_i^{t_2} - u_i^{t_2})^2} \leq n \right\}. \tag{3.29}$$

Notice that one of the two rectangular sets $\mathcal{T}_{t_1}$ and $\mathcal{T}_{t_2}$ may have no interior in the set $\mathcal{F}_\Delta$. In order to guarantee that the ellipsoids $\mathcal{E}_{t_1}$ and $\mathcal{E}_{t_2}$ have interior points in the set $\mathcal{F}_\Delta$, such rectangular set should be eliminated from further consideration. The following conditions can be used to determine which rectangular set should be eliminated:

$$e^T u^{t_1} < 1, \quad e^T v^{t_1} > 1 \tag{3.30}$$

and

$$e^T u^{t_2} < 1, \quad e^T v^{t_2} > 1 \tag{3.31}$$

If (3.30) is violated, then $\mathcal{T}_{t_1}$ should be eliminated. Moreover, $\mathcal{T}_t$ is replaced by $\mathcal{T}_{t_2}$ and $\mathcal{E}_t$ is replaced by $\mathcal{E}_{t_2}$; if (3.31) is violated, then $\mathcal{T}_{t_2}$ should be eliminated. Moreover, $\mathcal{T}_t$ is replaced by $\mathcal{T}_{t_1}$ and $\mathcal{E}_t$ is replaced by $\mathcal{E}_{t_1}$. Otherwise, $\mathcal{T}_t$ is replaced by $\mathcal{T}_{t_1}$ and $\mathcal{T}_{t_2}$ and $\mathcal{E}_t$ is replaced by $\mathcal{E}_{t_1}$ and $\mathcal{E}_{t_2}$.

After that, a point $\tilde{x} \in \mathbb{R}^n_+$ is obtained by setting all the negative components of x^* to be 0. If $\tilde{x}^T M \tilde{x} < 0$, then a violating vector $\tilde{x}$ is found and the algorithm halts. Otherwise the algorithm iterates until some stopping criterion is met. The proposed algorithm is presented below.

Adaptive Ellipsoid-based Algorithm for Detecting Copositivity (AEA-DC)

Initialization: Let $\mathcal{E}_1 = \{x \in \mathbb{R}^n | \sum_{i=1}^n (2x_i - 1)^2 \leq n\}$ and $\mathcal{T}_1 = [u^1, \ v^1]$, where $u_i^1 = 0, v_i^1 = 1$ for $i = 1, 2, \ldots, n$. Set $\mathbf{E} = \{\mathcal{E}_1\}$, $\mathbf{T} = \{\mathcal{T}_1\}$ and $\overline{\mathbf{T}} = \emptyset$. Set $\epsilon > 0$ to be the tolerance. Let l denote the best lower bound and s the best upper bound.

Step 1: Let $\mathscr{E} = \bigcup_{\mathcal{E}_i \in \mathbf{E}} \mathcal{E}_i$. Solve the problem (RCP-StQP) with the approximation cone $\mathscr{N}^*_{\mathscr{E}}$. Assume the optimal solution to problem (RCP-StQP) is $Y^* = \sum_{i:(Y^i)^* \neq 0} (Y^i)^*$. Return the optimal value of problem (RCP-StQP) as V(RCP-StQP). Set $l = \max\{l, V(\text{RCP-StQP})\}$. If $0 > l \geq -\epsilon$, then M is ϵ-copositive. Stop. If $l \geq 0$, then M is copositive. Stop. Otherwise, go to *Step 2*.

Step 2: Decompose Y^* according to Corollary 3.5.1 to obtain the most sensitive point x^* and the most sensitive ellipsoid $\mathcal{E}_t = \{x \in \mathbb{R}^n | \sum_{i=1}^n \frac{(2x_i - v_i^t - u_i^t)^2}{(v_i^t - u_i^t)^2} \leq n\} \in \mathbf{E}$, which is generated from the rectangular set $\mathcal{T}_t = [u^t,\ v^t] \in \mathbf{T}$. If $x^* \in \mathbb{R}^n_+$, then $(x^*)^T M x^* \leq l < -\epsilon$ and M is not copositive with a violating vector x^* being found. Stop. Otherwise, go to *Step 3.*

Step 3: Set $\mathbf{E} \leftarrow \mathbf{E} \setminus \{\mathcal{E}_t\}$ and $\mathbf{T} \leftarrow \mathbf{T} \setminus \{\mathcal{T}_t\}$. Define ellipsoids $\mathcal{E}_{t_1}$ and $\mathcal{E}_{t_2}$ according to (3.28) and (3.29). If (3.30) is violated, set $\mathbf{E} \leftarrow \mathbf{E} \cup \{\mathcal{F}_{t_2}\}$, $\mathbf{T} \leftarrow \mathbf{T} \cup \{\mathcal{T}_{t_2}\}$ and $\overline{\mathbf{T}} \leftarrow \overline{\mathbf{T}} \cup \{\mathcal{T}_{t_1}\}$. If (3.31) is violated, set $\mathbf{E} \leftarrow \mathbf{E} \cup \{\mathcal{E}_{t_1}\}$, $\mathbf{T} \leftarrow \mathbf{T} \cup \{\mathcal{T}_{t_1}\}$ and $\overline{\mathbf{T}} \leftarrow \overline{\mathbf{T}} \cup \{\mathcal{T}_{t_2}\}$. Otherwise, set $\mathbf{E} \leftarrow \mathbf{E} \cup \{\mathcal{E}_{t_1}\} \cup \{\mathcal{E}_{t_2}\}$ and $\mathbf{T} \leftarrow \mathbf{T} \cup \{\mathcal{T}_{t_1}\} \cup \{\mathcal{T}_{t_2}\}$.

Step 4: Generate a point $\tilde{x} \in \mathbb{R}^n_+$ by setting the negative components of x^* to 0. Set $s = \min\{s,\ \tilde{x}^T M \tilde{x}\}$. If $s < 0$, then M is not copositive with a violating vector $\tilde{x}$ found. Stop. Otherwise, go to *Step 1.*

Remark 1. In each iteration, the total volume of all the rectangles in $\mathbf{T}$ and $\overline{\mathbf{T}}$ is always 1. The set $\overline{\mathbf{T}}$ in the proposed algorithm is used for the convenience of the proof in Lemma 3.5.1. In a practical algorithm implementation, $\overline{\mathbf{T}}$ need not be stored.

In the proposed algorithm, at most one additional ellipsoid is added into the set $\mathscr{E}$ in each iteration. Thus, the complexity of solving problem (RCP-StQP) does not increase dramatically in each iteration. In order to prove the finite termination of the proposed algorithm, the following lemma is needed.

Lemma 3.5.1. *For any given $\delta > 0$, there exists an $N_\delta \in \mathbb{N}$ such that $\|x^* - \tilde{x}\|_\infty < \delta$ at the N_δ-th iteration.*

Proof. If $x^* \in \mathbb{R}^n_+$ happens at some N_0-th iteration, then $\tilde{x} = x^*$, and the lemma holds trivially. Otherwise, $x^* \notin \mathbb{R}^n_+$ at each iteration. Note that if $x^* \in \mathcal{B}_{\delta,\mathbb{R}^n_+} = \{x \in \mathbb{R}^n | \exists y \in \mathbb{R}^n_+,\ \text{s.t.}\ \|x - y\|_\infty < \delta\}$, then $\|x^* - \tilde{x}\|_\infty < \delta$. Now, we need to show that at some iteration, the most sensitive point x^* falls into $\mathcal{B}_{\delta,\mathbb{R}^n_+}$. By our arrangement of the ellipsoids, we know that the length of the i-th half axis of ellipsoid $\mathcal{E}_t$ is equal to $\frac{\sqrt{n}}{2}(v_i^t - u_i^t)$. Therefore, after $(\lceil \frac{\sqrt{n}}{\delta} \rceil)^n$ iterations, there exists at least some v and s that satisfy $v_i^t - u_i^t < \frac{\delta}{\sqrt{n}}$, for some $i \in \{1, 2, ..., n\}$ and $t \in \{1, 2, ..., k\}$. Otherwise, the total volume of all the generated rectangular sets in $\mathbf{T}$ and $\overline{\mathbf{T}}$ becomes greater than one. Note at this time, the length of the i-th half axis of the ellipsoid $\mathcal{E}_t$ generated by $[u^t,\ v^t]$ is less than $\frac{\delta}{2}$. Assume that this ellipsoid was generated at the N_1-th iteration. Then among the first N_1 iterations, there exists one iteration N_δ such that a rectangular set is split along the i-th direction with $v_i - u_i < \frac{2\delta}{\sqrt{n}}$. Thus the length of the i-th half-axis of that most sensitive ellipsoid is less than δ and, consequently, $x^* \in \mathcal{B}_{\delta,\mathbb{R}^n_+}$. □

Theorem 3.5.3. *For a given $\epsilon > 0$, there exists some $N_\epsilon \in \mathbb{N}$ such that $s - l < \epsilon$ at the N_ϵ-th iteration.*

Proof. For any $\epsilon > 0$, there exists some $\delta > 0$ such that $|\tilde{x}^T M \tilde{x} - (x^*)^T M x^*| < \epsilon$ for any $\|x^* - \tilde{x}\|_\infty < \delta$. According to Lemma 3.5.1, there exists some $N_\epsilon \in \mathbb{N}$ such that $\|x^* - \tilde{x}\|_\infty < \delta$. Since l and s record the best lower and upper bound values, respectively, we have $s - l \leq \tilde{x}^T M \tilde{x} - (x^*)^T M x^* < \epsilon$. □

Because either $l \geq -\epsilon$ or $s < 0$ holds when $s - l \leq \epsilon$, Theorem 3.5.3 indicates that the proposed algorithm will eventually stop at *Step 1*, *Step 2* or *Step 4* for any given tolerance $\epsilon > 0$. This leads to the following theorem:

Theorem 3.5.4. *For any given $\epsilon > 0$, the proposed adaptive ellipsoid-based algorithm for detecting copositivity terminates in a finite number of iterations. If the proposed algorithm stops at Step 1, then M is ϵ-copositive; if the proposed algorithm stops at Step 2 or Step 4, then M is noncopositive.*

3.5.3 Improving the lower bounds by RLT

The lower bounds obtained by the proposed algorithm can be further improved using the so-called RLT (Reformulation-Linearization Technique). Anstreicher [11] used the RLT-based inequalities to improve the SDP relaxations for QCQP. Note that the standard simplex $\mathcal{F}_\Delta$ is contained in the first orthant $\mathbb{R}^n_+$, hence the inequalities $Y \geq 0$ can be added into problem (RCP-StQP) in order to further improve the lower bound. The new relaxation problem is written as below.

$$
\text{(RLT-StQP)} \qquad \begin{aligned} \min \quad & H \bullet Y \\ & \begin{bmatrix} 2 & -e^T \\ -e & 0 \end{bmatrix} \bullet Y = 0,\ Y_{11} = 1, \\ & Y = Y^1 + Y^2 + \cdots + Y^k \geq 0, \\ & \begin{bmatrix} c^i & (b^i)^T \\ b^i & A^i \end{bmatrix} \bullet Y^i \leq 0,\ Y^i \in \mathcal{S}^{n+1}_+,\ i = 1, 2, \ldots, k. \end{aligned} \tag{3.32}
$$

Consequently, we may solve problem (RLT-StQP) instead of problem (RCP-StQP) in the proposed algorithm.

3.6 Numerical Examples

The algorithm has been implemented using MATLAB 7.14.0 on a computer with Intel Core 2 CPU 2.40Ghz and 3G memory. We also used SeDuMi 1.3 [116] to solve the problem (RLT-StQP). Our source code is available at <http://www.ise.ncsu.edu/fangroup/>. The tolerance ϵ is set to be 0.001 for all the numerical experiments.

Bundfuss and Dür [32] designed inner and outer polyhedral approximations for the copositive cone. The approximation cone $\mathcal{N}_{\mathcal{E}}$ we used is nonpolyhedral and could be more efficient in approximating copositive cone, which is also nonpolyhedral. For example, consider Example 4.1 of [32] with $M = \begin{bmatrix} 1/3 & -2/3 \\ -2/3 & 4/3 \end{bmatrix}$ being the optimal solution. Since it sits on the curved surface of $\mathcal{C}_3$, our algorithm took only one iteration to stop and claim the copositivity of M. Bundfuss and Dür [32] also considered two examples from [23], which are equivalent to determine whether the following two matrices $M_1 = Q_1 - \lambda_1 E$ and $M_2 = Q_2 - \lambda_2 E$, where

$$Q_1 = \begin{bmatrix} -14 & -15 & -16 & 0 & 0 \\ -15 & -14 & -12.5 & -22.5 & -15 \\ -16 & -12.5 & -10 & -26.5 & -16 \\ 0 & -22.5 & -26.5 & 0 & 0 \\ 0 & -15 & -16 & 0 & -14 \end{bmatrix},$$

$$Q_2 = \begin{bmatrix} 0.9044 & 0.1054 & 0.5140 & 0.3322 & 0 \\ 0.1054 & 0.8715 & 0.7385 & 0.5866 & 0.9751 \\ 0.5140 & 0.7385 & 0.6936 & 0.5368 & 0.8086 \\ 0.3322 & 0.5866 & 0.5368 & 0.5633 & 0.7478 \\ 0 & 0.9751 & 0.8086 & 0.7478 & 1.2932 \end{bmatrix},$$

$\lambda_1 = -16\frac{1}{3}$, $\lambda_2 = 0.4839$ and E is the all ones matrix, are copositive or not. For matrix M_1, our algorithm took 200 iterations (about 805 seconds) to obtain a lower bound of $-9.6922 \times 10^{-4} > \epsilon$, and hence claimed M_1 is ϵ-copositive. For matrix M_2, our algorithm took 8 iterations (about 0.80 seconds) to obtain a lower bound of $-7.2522 \times 10^{-4} > \epsilon$ and claimed M_2 is ϵ-copositive too. Note that M_1 is highly symmetric and sits on the boundary of $\mathcal{C}_5$, it is not easy to detect its copositivity. Similar behavior has been observed in applying other methods (see [25] and [131])

In order to show that our algorithm is efficient for general matrices, we tested matrices with different sizes from 10 to 80. For each size, 100 symmetric matrices were randomly generated with elements being uniformly distributed over $[-1, 1]$. Minimal, maximal, average iterations and average CPU time (in seconds) in terms of matrix size are reported in Table 3.1.

By using the adaptive ellipsoid-based algorithm for detecting copositivity, all randomly generated matrices were correctly detected to be noncopositive in the first iteration.

In [123], the following empirical test has been proposed: for each n =3, 4, 5, 6, 7, 8, 9, 10, one thousand symmetric matrices of order n with the diagonal elements being 1 and off-diagonal elements falling in $[-1, 1]$ are randomly generated. Their results showed that, for $n \geq 8$, the copositivity of some matrices was undetermined. Using our proposed algorithm, *all* matrices

Table 3.1: Results of random test for *AEA-DC*

Matrix size	Min.	Max.	Ave.	CPU time (sec.)
10	1	1	1	0.0675
20	1	1	1	0.2143
30	1	1	1	0.9382
40	1	1	1	3.4871
50	1	1	1	15.8762
60	1	1	1	82.9367
70	1	1	1	258.1257
80	1	1	1	722.8373

were successfully detected to be copositive or noncopositive in less CPU time. We also extend the experiments for $n = 20, 40, 60$ but with 100 matrices generated for each n. The results are reported in Table 3.2, in which, "# of unsolved problems" denotes the number of matrices whose copositivity is undetermined, "CPU time" denotes the average CPU time in second and "avg.iter." denotes the average number of iterations taken by the proposed adaptive ellipsoid-based algorithm for detecting copositivity. We also ran a simulation test for our algorithm

Table 3.2: Simulation tests comparing with Yang et al. [123]

Matrix	# of unsolved problems		CPU time (sec.)		Avg.iter.
Size	*AEA-DC*	[123]	*AEA-DC*	[123]	*AEA-DC*
3	0	0	0.0508	0.1	1.1650
4	0	0	0.1060	0.7	1.5670
5	0	0	0.3266	3.2	2.1960
6	0	0	1.2321	19.1	3.0140
7	0	0	3.1489	96.4	3.4600
8	0	8	1.8644	398.7	2.7710
9	0	6	2.0343	351.2	2.0040
10	0	2	0.8344	363.5	1.4200
20	0	-	0.5465	-	1.9500
40	0	-	4.2022	-	1.0000
60	0	-	101.7502	-	1.0000

with randomly generated copositive $n \times n$ matrices of the form $M = P + N$, where P is a positive semidefinite matrix and N is a matrix with no negative elements (100 matrices for each $n \in \{10, 20, 30, 40, 50, 60, 70\}$). All generated matrices were successfully detected to be copositive in the first few iterations. The main results for this simulation test are summarized in Table 3.3. In this table, “Min.”, “Max.”, and “Avg.” denote the minimal, maximal and average number of iterations, respectively. Moreover,“CPU time” means the average CPU time in second. The fast copositivity detection in the above simulation test is mainly due to the fact

Table 3.3: Results for simulation test of the form $P + N$.

Matrix Size	Iterations Min.	Max.	Avg.	CPU time (seconds)
10	1	4	1.1300	0.0823
20	1	22	1.7200	0.7541
30	1	9	1.7500	2.1752
40	1	6	1.9400	8.4304
50	1	17	2.1600	114.1480
60	1	32	3.6400	722.4638
70	1	31	3.3100	3396.2696

that a randomly generated matrix is less likely to be on the boundary of the copositive cone. Under this circumstance, our algorithm can detect copositivity very efficiently.

3.7 Summary

In this chapter, we have developed conic reformulations and approximations for problem (StQP) with its application in detecting copositivity. A new algorithm has been proposed to determine the copositivity of a given symmetric matrix. The algorithm is based on solving a sequence of linear conic programming problems defined on the dual cone of nonnegative quadratic functions over an elliptic cover of the original feasible domain. By utilizing an adaptive scheme, the number of constraints in each problem involved does not increase dramatically. This feature not only saves memory storage, but also relieves the computational effort in each iteration. Moreover, a better ellipsoid arrangement can further improve the efficiency of the proposed algorithm. In the adaptive scheme, since the determination of the most sensitive point is based on the objective values of all sensitive points, the information of matrix data is embedded in the proposed algorithm. Therefore, we have developed an algorithm that does not depend on the matric structure and is somehow data-driven. Our work can be readily extended to solving copositive programs and detecting the copositivity of nonhomogeneous quadratic functions over the nonnegative orthant.

Chapter 4

Quadratic Programming Problems over Convex Quadratic Constraints

In Chapter 3, we studied the conic reformulation and approximation to quadratic programming problems over the standard simplex and developed an algorithm with an adaptive scheme embedded for detecting copositive matrices. In this chapter, we further extend the theory developed in Chapter 3 to solve nonconvex quadratic programming problems subject to several convex quadratic constraints. The similar conic reformulations and approximations are derived and an algorithm with a new adaptive scheme is proposed. Under some mild assumptions, the convergence and ϵ-optimality of the obtained solution are guaranteed.

4.1 Introduction

In this chapter, we study the quadratic programming problems over a set of convex quadratic constraints in the following form:

$$\text{(ETRS)} \qquad \begin{aligned} \min \quad & f(x) = x^T P^0 x + 2(q^0)^T x \\ \text{s.t.} \quad & x^T P^j x + 2(q^j)^T x + \gamma^j \leq 0, \quad j = 1, \ldots, m, \end{aligned} \tag{4.1}$$

where P^j is positive semidefinite for $j = 1, \ldots, m$, $q^j \in \mathbb{R}^n$ is a vector for $j = 0, 1, ..., m$ and $\gamma^j \in \mathbb{R}$ is a scalar for $j = 1, ..., m$. The problem is known as extended trust-region subproblem when $m \geq 2$ [36]. Denote the feasible domain of problem (ETRS) as $\Delta = \{x \in \mathbb{R}^n | x^T P^j x + 2(q^j)^T x + \gamma^j \leq 0,\ j = 1, 2, \ldots, m\}$. Problem (ETRS) arises from the analysis and relaxation of NP-hard combinatorial optimization problems [89]. Problem (ETRS) is NP-hard in general since P^0 in the objective function may not be positive semidefinite.

When $m = 1$, *i.e.*, there is only one convex quadratic constraint, problem (ETRS) becomes the classical trust-region subproblem (TRS) that appeared in [37]. Conn et al. introduced several methods for solving this problem in [39]. Also, it can be solved efficiently by using the semidefinite programming (SDP) techniques, provided that the feasible domain Δ has an interior point (see Appendix B in [28]).

When $m = 2$ and P^1, P^2 are positive definite, problem (ETRS) arises from applying the trust region method to solve single equality constrained nonlinear programs proposed by Celis et al. in [37]. Rendl and Wolkowicz presented a sequential quadratic programming method in [95] that made use of this problem. When $m = 2$ and problem (ETRS) satisfies the Slater condition with P^1 or P^2 being positive definite, Ye and Zhang introduced a parameterized problem in [125] and showed that by following the trajectory generated by the parameterized problem, one would reach the optimal solution of problem (ETRS). Most recently, Burer and Anstreicher considered the classical trust region problem with one extra full-dimensional ellipsoid constraint in [36], resulting in the "two trust-region subproblems" sharing the same form as problem (ETRS) with $m = 2$. They provided a new relaxation including some second-order-cone constraints that strengthen the usual SDP relaxation to achieve a narrower duality gap. But the global optimality of the solution is not guaranteed. For $m \geq 3$, Zheng et al. [128] provided lower bounds for problem (ETRS) based on the best difference of convex functions decomposition.

In this chapter, we solve problem (ETRS) with $m \geq 3$ under the assumption that the feasible domain Δ is bounded and it has a nonempty interior. The idea of this chapter is motivated by Lu et al.'s work [74], in which they presented a theoretical framework for solving QCQP problems and gave a specific algorithm for solving the nonconvex quadratic optimization problem with box constraints. Here, we extend their approach to solving the nonconvex quadratic optimization problem over a set of convex quadratic constraints. We first develop a conic formulation and approximation to the original problem, then the approximation is further improved by the reformulation-linearization technique (RLT). In order to get an ϵ-optimal solution, an adaptive scheme is adopted to refine the approximation. One thing to point out is that the proposed algorithm only requires the feasible domain to be bounded and has a nonempty interior. Our work may also be applied to an extended trust-region subproblem with many linear and ellipsoid constraints, which arises from the relaxation of NP-hard combinatorial optimization problems [89].

4.2 Conic Reformulation and Approximation to Problem (ETRS)

According to Section 2.2, under the assumption that the feasible domain Δ is bounded with a nonempty interior, problem (ETRS) is equivalent to the following linear conic programming problem:

$$\text{(CP-ETRS)} \qquad \begin{array}{ll} \min & H^0 \bullet Y \\ \text{s.t.} & Y_{11} = 1, \\ & Y \in \mathscr{N}_\Delta^* \end{array} \tag{4.2}$$

where $H^0 = \begin{bmatrix} 0 & (q^0)^T \\ q^0 & P^0 \end{bmatrix} \in \mathcal{S}^{n+1}$ and

$$\mathscr{N}_\Delta^* = \texttt{cone} \left\{ Y \in \mathcal{S}^{n+1} \middle| Y = \begin{bmatrix} 1 \\ x \end{bmatrix} \begin{bmatrix} 1 \\ x \end{bmatrix}^T \text{ for some } x \in \Delta \right\} \tag{4.3}$$

as introduced in Section 3.2. The matrix in cone $\mathscr{N}_\Delta^*$ has a rank-one decomposition as in Theorem 2.3.1 since Δ is bounded and closed. The dual cone of $\mathscr{N}_\Delta^*$ is the cone of nonnegative quadratic functions over Δ, that is,

$$\mathscr{N}_\Delta = \left\{ U \in \mathcal{S}^{n+1} \middle| \begin{bmatrix} 1 \\ x \end{bmatrix}^T U \begin{bmatrix} 1 \\ x \end{bmatrix} \geq 0 \text{ for all } x \in \Delta \right\}. \tag{4.4}$$

Then, the linear conic dual problem of (CP-ETRS) is defined as

$$\text{(CD-ETRS)} \qquad \begin{array}{ll} \max & \sigma \\ \text{s.t.} & \begin{bmatrix} -\sigma & (q^0)^T \\ q^0 & P^0 \end{bmatrix} \in \mathscr{N}_\Delta, \quad \sigma \in \mathbb{R}. \end{array} \tag{4.5}$$

Although, under the assumption that Δ has a nonempty interior, both cones $\mathscr{N}_\Delta$ and $\mathscr{N}_\Delta^*$ are proper according to Theorem 3.2.2, there is no known efficient algorithm for computing the rank-one decomposition of matrices in $\mathscr{N}_\Delta^*$ as in Section 2.3. Therefore, similar to Section 3.3.2, we could use the dual cone of the cone of nonnegative quadratic functions over an elliptic cover of Δ to approximate the cone $\mathscr{N}_\Delta^*$. Let $\mathbf{E}$ be a collection of full-dimensional ellipsoids $\mathcal{E}_i$, *i.e.*

$$\mathbf{E} = \bigcup_{i=1}^{k} \{\mathcal{E}_i\}, \tag{4.6}$$

where

$$\mathcal{E}_i = \{x \in \mathbb{R}^n | x^T A^i x + 2(b^i)^T x + c^i \leq 0\} \tag{4.7}$$

with $A^i \in \mathcal{S}^n_{++}$, $b \in \mathbb{R}^n$ and $c^i \in \mathbb{R}$ for $i = 1, \ldots, k$. Let $\mathscr{E}$ be the union of ellipsoids in $\mathbf{E}$. We say $\mathscr{E}$ is an *elliptic cover* of Δ if

$$\Delta \subseteq \mathscr{E} = \bigcup_{i:\mathcal{E}_i \in \mathbf{E}} \mathcal{E}_i, \tag{4.8}$$

where each $\mathcal{E}_i$, for $i = 1, ..., k$, is of full-dimension defined in (4.7). Then, the cone $\mathscr{N}^*_{\mathscr{E}}$ has the same LMI representation as in equation (3.16) and there are efficient algorithms to decompose the matrices in this cone as shown in Corollary 3.5.1. Before relaxing the problem (CP-ETRS), we rewrite it as the following form:

$$\text{(CP-ETRS)} \qquad \begin{array}{ll} \min & H^0 \bullet Y \\ \text{s.t.} & H^j \bullet Y \leq 0, \quad j = 1, 2, ..., m, \\ & Y_{11} = 1, \\ & Y \in \mathscr{N}^*_{\Delta} \end{array} \tag{4.9}$$

by adding the redundant constraints $H^j \bullet Y \leq 0$, where $H^j = \begin{bmatrix} \gamma^j & (q^j)^T \\ q^j & P^j \end{bmatrix}$ for $j = 1, 2, ..., m$. To verify these constraints are redundant, notice that any nonzero matrix $Y \in \mathscr{N}^*_{\Delta}$, by the definition of $\mathscr{N}^*_{\Delta}$, can be decomposed into $Y = \sum_{i=1}^r \alpha_i y^i (y^i)^T$, where $\alpha_i > 0$, $y^i = \left[\begin{smallmatrix} 1 \\ x^i \end{smallmatrix}\right]$ with $x^i \in \Delta$ for some $r \in \mathbb{N}$ and $i = 1, 2, \ldots, r$. Therefore, $H^j \bullet Y = \sum_{i=1}^r \alpha_i (y^i)^T H^j y^i \leq 0$ for $j = 1, 2, ..., m$. But these redundant constraints may not be redundant anymore in the following relaxed linear conic programming problem:

$$\text{(RCP-ETRS)} \qquad \begin{array}{ll} \min & H^0 \bullet Y \\ \text{s.t.} & H^j \bullet Y \leq 0, \; j = 1, 2, ..., m, \\ & Y_{11} = 1, \\ & Y \in \mathscr{N}^*_{\mathscr{E}}, \end{array} \tag{4.10}$$

and its dual problem is:

$$\text{(RCD-ETRS)} \qquad \begin{array}{ll} \max & \sigma \\ \text{s.t.} & H^0 + \begin{bmatrix} -\sigma & 0 \\ 0 & 0 \end{bmatrix} + \sum_{j=1}^m \mu_j H^j \in \mathscr{N}_{\mathscr{E}}, \\ & \sigma \in \mathbb{R}, \; \mu_j \geq 0, \; j = 1, 2, ..., m. \end{array} \tag{4.11}$$

According to Corollary 3.3.4, problems (RCP-ETRS) and (RCD-ETRS) can be specifically rewritten as

$$
\text{(RCP-ETRS)} \qquad \begin{array}{ll} \min & H^0 \bullet Y \\ \text{s.t.} & H^j \bullet Y \le 0, \quad j = 1, 2, ..., m, \\ & Y = Y^1 + \cdots + Y^k, \; Y_{11} = 1, \\ & \begin{bmatrix} c^i & (b^i)^T \\ b^i & A^i \end{bmatrix} \bullet Y^i \le 0, \; Y^i \in \mathcal{S}_+^{n+1}, \quad i = 1, \cdots, k, \end{array} \tag{4.12}
$$

and

$$
\text{(RCD-ETRS)} \qquad \begin{array}{ll} \max & \sigma \\ \text{s.t.} & S = H^0 + \begin{bmatrix} -\sigma & 0 \\ 0 & 0 \end{bmatrix} + \sum_{j=1}^m \mu_j H^j, \\ & S + \lambda_i \begin{bmatrix} c^i & (b^i)^T \\ b^i & A^i \end{bmatrix} \in \mathcal{S}_+^{n+1} \quad i = 1, 2, \ldots, k, \\ & \lambda_i \ge 0, \; i = 1, 2, ..., k, \\ & \sigma \in \mathbb{R}, \; \mu_j \ge 0, \quad j = 1, 2, ..., m. \end{array} \tag{4.13}
$$

Moreover, a tighter lower bound for problem (CP-ETRS) could be obtained by applying the reformulation-linearization technique (RLT) to problem (RCP-ETRS). It results in the following problem:

$$
\text{(RLT-ETRS)} \qquad \begin{array}{ll} \min & H^0 \bullet Y \\ \text{s.t.} & Y = Y^1 + \cdots + Y^k, \; Y_{11} = 1, \\ & H^j \bullet Y^i \le 0 \quad j = 1, ..., m, \; i = 1, ..., k, \\ & \begin{bmatrix} c^i & (b^i)^T \\ b^i & A^i \end{bmatrix} \bullet Y^i \le 0, \quad Y^i \in \mathcal{S}_+^{n+1} \quad i = 1, ..., k, \end{array} \tag{4.14}
$$

by adding RLT-constraints $H^j \bullet Y^i \le 0$ for $j = 1, ..., m$ and $i = 1, ..., k$. Then, its dual problem becomes:

$$
\text{(DRLT-ETRS)} \qquad \begin{array}{ll} \max & \sigma \\ \text{s.t.} & S^i = H^0 + \begin{bmatrix} -\sigma & 0 \\ 0 & 0 \end{bmatrix} + \sum_{j=1}^m \mu_{ij} H^j, \\ & S^i + \lambda_i \begin{bmatrix} c^i & (b^i)^T \\ b^i & A^i \end{bmatrix} \in \mathcal{S}_+^{n+1} \quad i = 1, ..., k, \\ & \sigma \in \mathbb{R}, \quad \lambda_i \ge 0, \quad i = 1, ..., k, \\ & \mu_{ij} \ge 0, \quad i = 1, ..., k, \; j = 1, ..., m. \end{array} \tag{4.15}
$$

The next theorem shows that problem (RLT-ETRS) indeed provides a lower bound for problem (CP-ETRS) if $\Delta \subseteq \mathbf{E}$.

Theorem 4.2.1. *Let $\mathscr{E}$ and $\mathcal{E}_i$ be defined in* (4.8) and (3.11), respectively. If the set $\Delta \subseteq \mathscr{E}$, then V(RCP-ETRS)$\leq V$(RLT-ETRS)$\leq V$(CP-ETRS)$= V$(ETRS).

Proof. We only need to prove V(RLT-ETRS)$\leq V$(ETRS). For any feasible solution $x \in \Delta$, there exists some $i_0 \in \{1, ..., k\}$, such that $x \in \mathcal{E}_i$ due to the fact $\Delta \subseteq \mathbf{E}$. Let $Y^{i_0} = \begin{bmatrix} 1 \\ x \end{bmatrix} \begin{bmatrix} 1 \\ x \end{bmatrix}^T$, $Y^i = 0$ for $i \neq i_0$ and $Y = \sum_{i=1}^k Y^i$, then $(Y, Y^1, ..., Y^k)$ is feasible to problem (RLT-ETRS), thus V(RLT-ETRS)$\leq V$(ETRS). □

Under some mild assumptions, problems (RLT-ETRS) and (DRLT-ETRS) are both strictly feasible.

Theorem 4.2.2. *Let $\mathcal{E}_i$ be defined in* (3.11)*. If the set $\mathcal{E}_i \cap \Delta$ has a nonempty interior for $i = 1, \ldots, k$, then problem* (RLT-ETRS) *is strictly feasible. Moreover, problem* (DRLT-ETRS) *is always strictly feasible.*

Proof. Because the interior of set $\mathcal{E}_i \cap \Delta$ is nonempty, there is a point $\bar{x}^i$ such that $(\bar{x}^i)^T A^i \bar{x}^i + 2(b^i)^T \bar{x}^i + c^i < 0$ and $(\bar{x}^i)^T P^j \bar{x}^i + 2(q^j)^T \bar{x}^i + \gamma^j < 0$ for $j = 1, 2, ..., m$. Because $\bar{x}^i$ is an interior point of the set $\mathcal{E}_i \cap \Delta$, there exists an n-dimensional simplex with affinely independent vertices $\bar{x}^{is}$, $s = 1, \ldots, n+1$, contained in the interior of the set $\mathcal{E}_i \cap \Delta$ such that $\bar{x}^i$ is an interior point of this simplex. Then $\bar{x}^i = \sum_{s=1}^{n+1} \bar{\alpha}_{is} \bar{x}^{is}$ with $\bar{\alpha}_{is} > 0$ for $s = 1, \ldots, n+1$ and $\sum_{s=1}^{n+1} \bar{\alpha}_{is} = 1$. Consider the matrix

$$\bar{Y}^i = \frac{1}{k} \sum_{s=1}^{n+1} \bar{\alpha}_{is} \begin{bmatrix} 1 \\ \bar{x}^{is} \end{bmatrix} \begin{bmatrix} 1 \\ \bar{x}^{is} \end{bmatrix}^T$$

for $i = 1, \ldots, k$. It is easy to check that $\begin{bmatrix} c^i & (b^i)^T \\ b^i & A^i \end{bmatrix} \bullet \bar{Y}^i < 0$ and $H^j \bullet \bar{Y}^i < 0$ for $j = 1, ..., m$. Also, $\bar{Y}^i \in \mathcal{S}_{++}^{n+1}$ because the vertices $\bar{x}^{is}$, $s = 1, ..., n+1$, are affinely independent. Let $\bar{Y} = \sum_{i=1}^k \bar{Y}^i$. Then $(\bar{Y}^1, \ldots, \bar{Y}^k, \bar{Y})$ is a strictly feasible solution of problem (RLT-ETRS). This proves that problem (RLT-ETRS) is strictly feasible.

For any $\bar{\mu}_{ij} > 0$, $i = 1, ..., k$ and $j = 1, ..., m$, in problem (DRLT-ETRS), there exists a sufficiently large $\bar{\lambda}_i > 0$ for $i = 1, ..., k$ and a sufficiently small $\bar{\sigma}$ such that the matrix

$$\bar{S}^i = H^0 + \begin{bmatrix} -\bar{\sigma} & 0 \\ 0 & 0 \end{bmatrix} + \sum_{j=1}^m \bar{\mu}_{ij} H^j$$

satisfies

$$\bar{S}^i + \bar{\lambda}_i \begin{bmatrix} c^i & (b^i)^T \\ b^i & A^i \end{bmatrix} \in \mathcal{S}^{n+1}_{++} \text{ for } i = 1, 2, \ldots, k$$

by diagonal dominance. This proves the strong feasibility of problem (DRLT-ETRS). □

By the linear duality theory of Theorem 2.4.2, we have the following corollary.

Corollary 4.2.3. *If the set $\mathcal{E}_i \cap \Delta$ has a nonempty interior for $i = 1, 2, ..., k$, then there is no duality gap between problems* (RLT-ETRS) *and* (DRLT-ETRS)*. Moreover, the optimal solutions of both problems are attainable.*

By the linear conic optimality conditions in Corollary 2.4.3, the optimality conditions between problems (RCP-ETRS) and (RCD-ETRS) are

$$\begin{aligned} &(Y^1, Y^2, \ldots, Y^k, Y) \text{ is feasible to problem (RLT-ETRS)}, \\ &(\sigma, \mu, \lambda, S^1, \ldots, S^k) \text{ is feasible to problem (DRLT-ETRS)}, \qquad \text{(Optimality Conditions)} \\ &\left(S^i + \lambda_i \begin{bmatrix} c^i & (b^i)^T \\ b^i & A^i \end{bmatrix} \right) \bullet Y^i = 0 \text{ for } i = 1, 2, \ldots, k. \end{aligned}$$

According to Theorem 4.2.1, the optimal value of problem (RLT-ETRS) gives a lower bound to problem (ETRS) when $\Delta \subseteq \mathscr{E}$. In order to further improve the lower bound, we could refine the elliptic cover $\mathscr{E}$ such that $\mathscr{E}$ becomes a better approximation to the feasible domain Δ of problem (ETRS). In the next section, we will show how to refine $\mathscr{E}$ in an adaptive way and prove that the lower bounds obtained by continuously refining $\mathscr{E}$ indeed converge to the optimal value of problem (ETRS).

4.3 An Adaptive Scheme for Problem (ETRS)

In order to refine the elliptic cover $\mathscr{E}$ in an efficient way, we need to know which ellipsoid $\mathcal{E}_i$ in $\mathscr{E}$ should be refined. As in Subsection 3.5.1, we need the definitions of *sensitive points* and *sensitive ellipsoids*. From Corollary 3.5.1, we know that if $Y^* = (Y^1)^* + \cdots + (Y^k)^*$ is the optimal solution to problem (RLT-ETRS), then

$$Y^* = \sum_{i:(Y^i)^* \neq 0} \sum_{s=1}^{n_i} \alpha_{is} \begin{bmatrix} 1 \\ x^{is} \end{bmatrix} \begin{bmatrix} 1 \\ x^{is} \end{bmatrix}^T \tag{4.16}$$

for some $n_i \in \{1, ..., n+1\}$, $\alpha_{is} > 0$, $x^{is} \in \mathcal{E}_i$ and $\sum_{i:(Y^i)^* \neq 0} \sum_{s=1}^{n_i} \alpha_{is} = 1$. We define all the points x^{is} in (4.16), for $i = 1, 2, ..., k$ and $s = 1, 2, ..., n_i$, to be *sensitive points*. Also, we define the ellipsoid $\mathcal{E}_i$ containing these x^{is} as a *sensitive ellipsoid*.

Among all these sensitive points, we define the *most sensitive point* in below.

Definition 6. For the rank-one decomposition

$$Y^* = \sum_{i:(Y^i)^* \neq 0} \sum_{s=1}^{n_i} \alpha_{is} \begin{bmatrix} 1 \\ x^{is} \end{bmatrix} \begin{bmatrix} 1 \\ x^{is} \end{bmatrix}^T, \tag{4.17}$$

x^* is the *most sensitive point* if

$$x^* = \operatorname{argmin}_{\{x^{is}:(Y^i)^* \neq 0;\ s=1,2,\ldots,n_i\}} \left\{ (x^{is})^T P^0 x^{is} + 2(q^0)^T x^{is} \right\}. \tag{4.18}$$

That is, x^* has the minimum objective value among all of the sensitive points. Note that there could be multiple sensitive points having the same minimum objective value. Under this case, we choose x^* as the one having the smallest index in i with the smallest index in s as a tie breaker. Denote this smallest index i by t. Then $\mathcal{E}_t$ is the ellipsoid containing x^* and $\mathcal{E}_t$ is the *most sensitive ellipsoid*.

Remark 2. Since the decomposition of $(Y^i)^*$ can be achieved in polynomial time (see [117] or [125] for the detailed procedure), thus the decomposition of Y^* in (4.17) can be obtained efficiently.

The same proof for Theorem 3.5.2 shows that the objective value at the most sensitive point is a lower bound of problem (ETRS). We state the result in the next theorem.

Theorem 4.3.1. *Assume Y^* is an optimal solution to problem* (RLT-ETRS) *with the most sensitive point x^*, then*

$$\begin{bmatrix} 1 \\ x^* \end{bmatrix} \begin{bmatrix} 1 \\ x^* \end{bmatrix}^T \bullet H^0 \leq V(\text{ETRS}). \tag{4.19}$$

Moreover, if $x^* \in \Delta$, then the matrix $\begin{bmatrix} 1 \\ x^* \end{bmatrix} \begin{bmatrix} 1 \\ x^* \end{bmatrix}^T$ *is optimal to problem* (CP-ETRS) *and x^* is optimal to problem* (ETRS).

In order to easily manage the ellipsoids in $\mathbf{E}$, each ellipsoid $\mathcal{E}_i$, as defined by (3.25), is generated from a rectangular set $\mathcal{T}_i = [u^i, v^i]$. We have the following lemma about the volume of these two sets. Let $\text{Vol}(S)$ be the volume of a set $S \subseteq \mathbb{R}^n$.

Lemma 4.3.1. *If $\mathcal{T} = [u,\ v]$ and $\mathcal{E}_{\mathcal{T}} = \left\{ x \in \mathbb{R}^n | \sum_{i=1}^n \frac{(2x_i - v_i - u_i)^2}{(v_i - u_i)^2} - n \leq 0 \right\}$, then* $\text{Vol}(\mathcal{E}_{\mathcal{T}}) = (\frac{n\pi}{4})^{n/2} \frac{1}{\Gamma(\frac{n}{2}+1)} \text{Vol}(\mathcal{T})$.

Proof. For an ellipsoid $\mathcal{E} = \{x \in \mathbb{R}^n | (x - c_0)^T Q(x - c_0) \leq 1\}$ with c_0 being the center and $Q \in \mathcal{S}^n_{++}$, $\mathrm{Vol}(\mathcal{E}) = \frac{\pi^{n/2}}{\Gamma(\frac{n}{2}+1)} \frac{1}{\sqrt{\det Q}}$ (ref. [55]). Therefore,

$$
\begin{aligned}
\mathrm{Vol}(\mathcal{E}_{\mathcal{T}}) &= \left(\frac{n\pi}{4}\right)^{n/2} \frac{1}{\Gamma(\frac{n}{2}+1)} \prod_{i=1}^{n} (v_i - u_i) \\
&= \left(\frac{n\pi}{4}\right)^{n/2} \frac{1}{\Gamma(\frac{n}{2}+1)} \mathrm{Vol}(\mathcal{T}).
\end{aligned}
$$

□

4.3.1 An adaptive scheme for (ETRS)

Similar to Section 3.5.2, let

$$
\mathbf{T} = \bigcup_{i=1}^{k} \{\mathcal{T}_i\}
$$

be a collection of rectangular sets $\mathcal{T}_i$ for $i = 1, ..., k$ and

$$
\Delta \subseteq \mathscr{T} = \bigcup_{i:\mathcal{T}_i \in \mathbf{T}} \mathcal{T}_i
$$

be the rectangular set cover of Δ. Then $\mathscr{E} = \mathcal{E}_1 \cup \cdots \cup \mathcal{E}_k$ is an elliptic cover of Δ, where each $\mathcal{E}_i$ is generated according to (3.25). In order to make our algorithm converge quickly, a good initial rectangular set cover is necessary. In order to fulfill this purpose, consider the following problems:

$$
(I^i_{\min}) \qquad \begin{array}{ll} \min & x_i \\ \text{s.t.} & x^T P^j x + 2(q^j)^T x + \gamma^j \leq 0, \quad j = 1, 2, \ldots, m, \end{array} \tag{4.20}
$$

and

$$
(I^i_{\max}) \qquad \begin{array}{ll} \max & x_i \\ \text{s.t.} & x^T P^j x + 2(q^j)^T x + \gamma^j \leq 0, \quad j = 1, 2, \ldots, m, \end{array} \tag{4.21}
$$

for $i = 1, \ldots, n$. Note that problem $(I^i_{\min})$ and problem $(I^i_{\max})$ are convex optimization problems, hence they can be solved efficiently. Also note that the feasible domain Δ is closed and bounded, problem $(I^i_{\min})$ and problem $(I^i_{\max})$ have finite optimal solutions. Denote the optimal solutions of $(I^i_{\min})$ and $(I^i_{\max})$ to be u^1_i and v^1_i. The rectangular set $\mathcal{T}_1 = [u^1,\ v^1]$ has the smallest volume among those rectangular sets covering the feasible domain Δ. This rectangular set is chosen as the initial rectangle set cover of the feasible domain Δ, and the ellipsoid $\mathcal{E}_1$ generated from $\mathcal{T}_1$ is the initial elliptic cover $\mathscr{E}$ of Δ. The existence of $\mathcal{T}_1$ is guaranteed by the assumption that the feasible domain Δ is bounded with a nonempty interior. Moreover, the set $\mathcal{T}_1 \cap \Delta$ has a nonempty interior.

When the most sensitive point x^* and the most sensitive ellipsoid $\mathcal{E}_t$ are detected, the rectangular set $\mathcal{T}_t$ generating the most sensitive ellipsoid $\mathcal{E}_t$ is also detected. Then, this rectangular set is split in half along the direction perpendicular to the longest edge. That is, if $\mathcal{T}_t = [u^t,\ v^t]$, with $id = \text{argmax}_{\{i=1,\dots,n\}}\{v_i^t - u_i^t\}$, then $\mathcal{T}_t$ is split into $\mathcal{T}_{t_1} = [u^{t_1},\ v^{t_1}]$ and $\mathcal{T}_{t_2} = [u^{t_2},\ v^{t_2}]$, where $u^{t_1} = u^t$, $v^{t_2} = v^t$, $v_i^{t_1} = v_i^t$, $u_i^{t_2} = u_i^t$, for $i \neq id$, and $v_{id}^{t_1} = u_{id}^{t_2} = \frac{u_{id}^t + v_{id}^t}{2}$. Two ellipsoids $\mathcal{E}_{t_1}$ and $\mathcal{E}_{t_2}$ are generated from $\mathcal{T}_{t_1}$ and $\mathcal{T}_{t_2}$ according to

$$\mathcal{E}_{t_1} = \left\{ x \in \mathbb{R}^n \,\middle|\, \sum_{i=1}^{n} \frac{(2x_i - v_i^{t_1} - u_i^{t_1})^2}{(v_i^{t_1} - u_i^{t_1})^2} \leq n \right\}, \tag{4.22}$$

and

$$\mathcal{E}_{t_2} = \left\{ x \in \mathbb{R}^n \,\middle|\, \sum_{i=1}^{n} \frac{(2x_i - v_i^{t_2} - u_i^{t_2})^2}{(v_i^{t_2} - u_i^{t_2})^2} \leq n \right\}. \tag{4.23}$$

If we manage the rectangular set cover $\mathscr{T}$ in such way that the set $\mathcal{T}_i \cap \Delta$ has a nonempty interior for each rectangular set $\mathcal{T}_i$ in $\mathscr{T}$, then at most one of the two rectangular sets $\mathcal{T}_{t_1}$ and $\mathcal{T}_{t_2}$ may have no common interior with the feasible domain Δ. If such rectangular set exists, it should be eliminated from the rectangular set cover $\mathscr{T}$ for further consideration. In order to determine which rectangular set should be eliminated, consider the following problems:

$$(I_{\min}^{id}) \qquad \begin{array}{ll} \min & x_{id} \\ \text{s.t.} & x^T P^j x + 2(q^j)^T x + \gamma^j \leq 0, \quad j = 1, 2, \dots, m, \\ & u^t \leq x \leq v^t \end{array} \tag{4.24}$$

and

$$(I_{\max}^{id}) \qquad \begin{array}{ll} \max & x_{id} \\ \text{s.t.} & x^T P^j x + 2(q^j)^T x + \gamma^j \leq 0, \quad j = 1, 2, \dots, m, \\ & u^t \leq x \leq v^t. \end{array} \tag{4.25}$$

Note that problem $(I_{\min}^{id})$ and problem $(I_{\max}^{id})$ are convex optimization problems (this is why the constraints need to be convex), hence they can be solved efficiently. Denote the optimal value of $(I_{\min}^{id})$ and $(I_{\max}^{id})$ to be ϕ and ψ, respectively. We manage the rectangular sets in $\mathbf{T}$ in the following way.

$$\mathbf{T} = (\mathbf{T} \setminus \{\mathcal{T}_t\}) \cup \{\mathcal{T}_{t_2}\}, \quad \text{if} \quad \phi \geq \frac{u_{id}^t + v_{id}^t}{2}; \tag{4.26}$$

$$\mathbf{T} = (\mathbf{T} \setminus \{\mathcal{T}_t\}) \cup \{\mathcal{T}_{t_1}\}, \quad \text{if} \quad \psi \leq \frac{u_{id}^t + v_{id}^t}{2}; \tag{4.27}$$

$$\mathbf{T} = (\mathbf{T} \setminus \{\mathcal{T}_t\}) \cup \{\mathcal{T}_{t_1}\} \cup \{\mathcal{T}_{t_2}\}, \quad \text{otherwise.} \tag{4.28}$$

The following theorem shows that if we manage rectangular set cover $\mathscr{T} = \cup_{i:\mathcal{T}_i \in \mathbf{T}} \mathcal{T}_i$ in this way, then each rectangular set $\mathcal{T}_i$ in $\mathbf{T}$ has a common interior with the feasible domain Δ.

Theorem 4.3.2. *The set $\mathcal{T}_i \cap \Delta$ has a nonempty interior for each rectangular set $\mathcal{T}_i$ in $\mathbf{T}$ if the rectangular sets are added into $\mathbf{T}$ according to* (4.26)-(4.28).

Proof. We only need to prove that the newly added rectangle set $\mathcal{T}_{t_1}$ or $\mathcal{T}_{t_2}$ based on (4.26)-(4.28) has a common interior with Δ. Note that $\mathcal{T}_t = \mathcal{T}_{t_1} \cup \mathcal{T}_{t_2}$ and $\mathcal{T}_t \cap \Delta$ has a nonempty interior, either $\mathcal{T}_{t_1} \cap \Delta$ or $\mathcal{T}_{t_2} \cap \Delta$ has a nonempty interior. If $\phi \geq \frac{u_{id}^t + v_{id}^t}{2}$, then the set $\mathcal{T}_{t_1} \cap \Delta$ has no interior, and should not be added into $\mathbf{T}$. In fact, we have $\mathcal{T}_t \cap \Delta = \mathcal{T}_{t_1} \cap \Delta$. To see this, assume $x^{t_1} \in (\mathcal{T}_t \setminus \mathcal{T}_{t_2}) \cap \Delta$, then $x_{id}^{t_1} < \frac{u_{id}^t + v_{id}^t}{2} \leq \phi$. But x^{t_1} is a feasible solution of problem $(I_{\min}^{id})$ defined by (4.24), which contradicts the fact that ϕ is optimal value. Similarly, when $\psi \leq \frac{u_{id}^t + v_{id}^t}{2}$, $\mathcal{T}_t \cap \Delta = \mathcal{T}_{t_1} \cap \Delta$ and the set $\mathcal{T}_{t_2} \cap \Delta$ has no interior point. When $\phi < \frac{u_{id}^{t_1} + v_{id}^{t_1}}{2} < \psi$, let x^{int} be an interior point of the set $\mathcal{T}_t \cap \Delta$, and let $x^{\min}$ and $x^{\max}$ be the optimal solutions of problems $(I_{\min}^{id})$ and $(I_{\max}^{id})$ defined by (4.24) and (4.25), respectively. Then the points $x^{t_1} = \lambda x^{\text{int}} + (1-\lambda) x^{\min}$ and $x^{t_2} = \eta x^{\text{int}} + (1-\eta) x^{\max}$ are interior points of the sets $\mathcal{T}_{t_1} \cap \Delta$ and $\mathcal{T}_{t_2} \cap \Delta$ for some $\lambda \in [0,\ 1)$ and $\eta \in [0,\ 1)$, respectively. (See Theorem 6.1 in [96].) □

Remark 3. The proof of Theorem 4.3.2 shows that $\mathscr{T}$ is still a rectangular set cover of Δ after refinement. Consequently, $\mathscr{E}$ is still an elliptic cover of Δ.

Since each $\mathcal{T}_i$ has a common interior with Δ, each ellipsoid $\mathcal{E}_i$, generated by $\mathcal{T}_i$, also has a common interior with Δ. Therefore, we have the following corollary:

Corollary 4.3.3. *The set $\mathcal{E}_i \cap \Delta$ has a nonempty interior for each $\mathcal{E}_i$ generated by the rectangular set $\mathcal{T}_i$ in $\mathbf{T}$.*

Notice that the assumptions in Theorem 4.2.2 and Corollary 4.2.3 hold if we manage the rectangular set cover by (4.26)-(4.28). According to Lemma 4.3.1, if the rectangular set $\mathcal{T}_t$ is split into two rectangular sets $\mathcal{T}_{t_1}$ and $\mathcal{T}_{t_2}$, then $\text{Vol}(\mathcal{E}_{t_1} \cup \mathcal{E}_{t_2}) < \text{Vol}(\mathcal{E}_t)$. Therefore, the volume of elliptic cover $\mathscr{E}$ is highly likely to strictly decrease after each refinement. This indicates that the elliptic cover $\mathscr{E}$ may become a better estimate of the feasible domain Δ.

At last, in order to check whether the current sensitive point x^* is close enough to the feasible domain Δ, consider the following problem (I^c):

$$(I^c) \qquad \begin{array}{ll} \min & \|x - x^*\|_\infty \\ \text{s.t.} & x^T P^j x + 2(q^j)^T x + \gamma^j \le 0, \quad j = 1, \dots, m. \end{array} \tag{4.29}$$

Again, (I^c) is a convex programming problem. Assume the optimal solution of (I^c) is $\bar{x}^1$. If $\|\bar{x}^1 - x^*\|_\infty \le \delta$, then $x^* \in B_{\delta,\Delta}$ and $\bar{x}^1$ could be output as an approximate optimal solution when δ is sufficiently small. Besides, we can get a feasible solution $\bar{x}^2$ easily from the optimal solution Y^* of problem (RLT-ETRS).

Lemma 4.3.2. *Assume $Y^* = (Y^1)^* + \cdots + (Y^k)^*$ is the optimal solution of problem* (RLT-ETRS) *and Y^* has a rank-one decomposition as in* (4.16)*. Then*

$$\bar{x}^2 = \sum_{i:(Y^i)^* \neq 0} \sum_{s=1}^{n_i} \alpha_{is} x^{is} \tag{4.30}$$

is a feasible solution of problem (ETRS)*.*

Proof. Since $Y^* = \begin{bmatrix} 1 & (\bar{x}^2)^T \\ \bar{x}^2 & X \end{bmatrix} \in \mathcal{S}_+^{n+1}$, then $X - \bar{x}^2(\bar{x}^2)^T \in \mathcal{S}_+^n$ using the Schur complementary condition. Note that $P^j \in \mathcal{S}_+^n$ and $H^j \bullet Y \le 0$, we have

$$(\bar{x}^2)^T P^j \bar{x}^2 + 2(q^j)^T \bar{x}^2 + \gamma^j \le P^j \bullet X + 2(q^j)^T \bar{x}^2 + \gamma^j = H^j \bullet Y \le 0.$$

Hence, $\bar{x}^2$ is a feasible solution of problem (ETRS). □

Our proposed algorithm is stated as below.

Adaptive Ellipsoid-based Algorithm for ETRS (AEA-ETRS)

Initialization : Solve problems $(I^i_{\min})$ and $(I^i_{\max})$ defined by (4.20) and (4.21) for $i = 1, ..., n$ to get the initial rectangle set $\mathcal{T}_1$ and the corresponding ellipsoid $\mathcal{E}_1$. Set $\epsilon > 0$ be the tolerance, $\mathbf{T} = \{\mathcal{T}_1\}$, $\mathbf{E} = \{\mathcal{E}_1\}$ and $\overline{\mathbf{T}} = \emptyset$. Let lower bound $l = -\infty$, upper bound $s = +\infty$ and approximate solution $\tilde{x} = 0 \in \mathbb{R}^n$.

Step 1 : Solve problem (RLT-ETRS) defined by (4.12) with the approximation cone $\mathscr{N}^*_{\mathscr{E}}$, where $\mathscr{E}$ is defined by (4.8). Assume the optimal solution to problem (RLT-ETRS) is $Y^* = (Y^1)^* + \cdots + (Y^k)^*$. Return the optimal value of problem (RLT-ETRS) as l^*. Set $l = \max\{l, l^*\}$. If $|s - l| \le \epsilon$, stop and output $\tilde{x}$. Otherwise, go to *Step 2*.

Step 2 : Decompose Y^* according to Corollary 3.5.1 to obtain the most sensitive point x^* and the most sensitive ellipsoid $\mathcal{E}_t = \{x \in \mathbb{R}^n | \sum_{i=1}^n \frac{(2x_i - v_i^t - u_i^t)^2}{(v_i^t - u_i^t)^2} \leq n\} \in \mathbf{E}$, which is generated from the rectangle set $\mathcal{T}_t = [u^t,\ v^t] \in \mathbf{T}$. If $x^* \in \Delta$, stop and output x^*. Otherwise, go to *Step 3*.

Step 3 : Set $\mathbf{T} = \mathbf{T} \setminus \{\mathcal{T}_t\}$, $\mathbf{E} = \mathbf{E} \setminus \{\mathcal{E}_t\}$ and $id = \text{argmax}_{\{i=1,\ldots,n\}}\{v_i^t - u_i^t\}$. Generate ellipsoids $\mathcal{E}_{t_1}$ and $\mathcal{E}_{t_2}$ according to (4.22) and (4.23), respectively. Let ϕ and ψ be the optimal value of problems $(I_{\min}^{id})$ and $(I_{\max}^{id})$ defined by (4.24) and (4.25), respectively.

- If $\phi \geq \frac{u_{id}^t + v_{id}^t}{2}$, set $\mathbf{E} = \mathbf{E} \cup \{\mathcal{E}_{t_2}\}$, $\mathbf{T} = \mathbf{T} \cup \{[u^{t_2},\ v^{t_2}]\}$ and $\overline{\mathbf{T}} = \overline{\mathbf{T}} \cup \{[u^{t_1},\ v^{t_1}]\}$;
- If $\psi \leq \frac{u_{id}^t + v_{id}^t}{2}$, set $\mathbf{E} = \mathbf{E} \cup \{\mathcal{E}_{t_1}\}$, $\mathbf{T} = \mathbf{T} \cup \{[u^{t_1},\ v^{t_1}]\}$ and $\overline{\mathbf{T}} = \overline{\mathbf{T}} \cup \{[u^{t_2},\ v^{t_2}]\}$;
- Otherwise, set $\mathbf{E} = \mathbf{E} \cup \{\mathcal{E}_{t_1}\} \cup \{\mathcal{E}_{t_2}\}$ and $\mathbf{T} = \mathbf{T} \cup \{[u^{t_1},\ v^{t_1}]\} \cup \{[u^{t_2},\ v^{t_2}]\}$.

Step 4 : Solve problem (I^c) defined by (4.29) to obtain $\bar{x}^1$. Set $\bar{x}^2 = \sum_{i:(Y^i)^* \neq 0} \sum_{s=1}^{n_i} \alpha_{is} x^{is}$. If $\min\{f(\bar{x}^1), f(\bar{x}^2)\} < s$, set $s = \min\{f(\bar{x}^1), f(\bar{x}^2)\}$ and

$$\tilde{x} = \text{argmin}_{\{\bar{x}^1, \bar{x}^2\}}\{f(\bar{x}^1), f(\bar{x}^2)\}.$$

If $|s - l| < \epsilon$, stop and output $\tilde{x}$. Otherwise, go to *Step 1*.

Remark 4. *Step 3* shows that at most one additional ellipsoid is added to the current ellipsoid cover at the end of each iteration. Therefore, the complexity of problem (RLT-ETRS) does not increase drastically.

Remark 5. The set $\overline{\mathbf{T}}$ in the proposed algorithm is used for the convenience of the proof in Lemma 4.3.3. In an actual algorithm implementation, $\overline{\mathbf{T}}$ need not be stored. Besides, the total volume of all rectangle sets in $\mathbf{T}$ and $\overline{\mathbf{T}}$ always equals to the volume of the initial rectangle set $\mathcal{T}_1$.

Remark 6. Two requirements need to be satisfied for the algorithm. One is that the feasible domain should be bounded with nonempty interior points such that the initial rectangle set can be found in the *Initialization* step and problem $(I_{\min}^{id})$ and problem $(I_{\max}^{id})$ are solvable in *Step 3*. The other requirement is that the objective function has to be quadratic such that the problem (RCP-ETRS) can be solved. This algorithm has some extensions to be seen in Section 4.6.

4.3.2 Proof of convergence

In this subsection, we show that the algorithm terminates in a finite number of iterations for any given $\epsilon > 0$ such that output objective value is within the given tolerance. The next lemma is useful in the proof.

Lemma 4.3.3. *For any given instance of problem* (ETRS) *and $\delta > 0$, there exists an $N_\delta \in \mathbb{N}$ such that $\|\bar{x} - x^*\|_\infty \leq \delta$ at the N_δ-th iteration.*

Proof. If at some N_0-th iteration, $x^* \in \Delta$, then $\bar{x} = x^*$, the lemma holds trivially. Otherwise, assume that $x^* \notin \Delta$ at each iteration. Let $B_{\delta,\Delta} = \{x \in \mathbb{R}^n | \exists y \in \Delta,\ s.t. \|x - y\|_\infty \leq \delta\}$. Note that if $x^* \in B_{\delta,\Delta}$, then $\|\bar{x} - x^*\|_\infty \leq \delta$. Therefore, we only need to show that at some iteration, the most sensitive point x^* falls into the set $B_{\delta,\Delta}$. Denote δ_1 as the longest edge of the initial rectangular set $\mathcal{T}_1$. By the arrangement of the ellipsoids, we know that the length of i-th half axis of ellipsoid $\mathcal{F}_t$ is equal to $\frac{\sqrt{n}}{2}(v_i^t - u_i^t)$, where $\mathcal{T}_t = [u^t,\ v^t]$. Then after $(\lceil \frac{\delta_1 \sqrt{n}}{\delta} \rceil)^n$ iterations, at least some rectangular set $[u,\ v]$ satisfies that $v_i - u_i \leq \frac{\delta}{\sqrt{n}}$ for some $i \in \{1, 2, ..., n\}$. Otherwise, the total volume of all rectangular sets in $\mathbf{T}$ and $\overline{\mathbf{T}}$ is greater than δ_1^n after $(\lceil \frac{\delta_1 \sqrt{n}}{\delta} \rceil)^n$ iterations. Note that the length of i-th half axis of the ellipsoid corresponding to rectangle $[u,\ v]$ at this time is less than $\frac{\delta}{2}$. Assume that this ellipsoid was generated at the N_1-th iteration. Then among the first N_1 iterations, there exists one iteration N_δ such that a rectangular set is split perpendicularly to some id-th direction with $v_{id} - u_{id} \leq \frac{2\delta}{\sqrt{n}}$. Thus the length of the id-th half-axis of the most sensitive ellipsoid is less than δ. Note that the id-th half-axis is the longest half-axis in the most sensitive ellipsoid and it intersects the feasible domain Δ, hence $x^* \in B_{\delta,\Delta}$. □

Remark 7. In the proof of Lemma 4.3.3, δ_1 depends on the given instance of problem (ETRS), so does N_δ. However, for any given instance, our proposed algorithm converges in finite iterations for a given tolerance.

Theorem 4.3.4. *Assume the feasible domain Δ of problem* (ETRS) *is bounded and has a nonempty interior. For any given instance with a tolerance $\epsilon > 0$, there exists an $N_\epsilon \in \mathbb{N}$ such that $|s - l| < \epsilon$ at the N_ϵ-th iteration of the proposed AEA-ETRS.*

Proof. For any $\epsilon > 0$, there exists some $\delta > 0$ such that $|f(\bar{x}) - f(x^*)| < \epsilon$ for any $\bar{x}$ and x^* satisfying $\|\bar{x} - x^*\|_\infty < \delta$, because the objective function $f(x)$ of problem (ETRS) is continuous on the compact set Δ. According to Lemma 4.3.3, there exists some $N_\epsilon \in \mathbb{N}$ such that $\|\bar{x} - x^*\|_\infty < \delta$. Since l and s record the best lower and upper bounds, we have $|s - l| \leq |f(\bar{x}) - f(x^*)| < \epsilon$. □

Theorem 4.3.4 shows that the solution $\tilde{x}$ obtained by our algorithm is an ϵ-optimal solution and our algorithm terminates in finite steps for any given instance with $\epsilon > 0$.

4.4 Numerical Examples

The proposed algorithm was implemented using MATLAB 7.14.0 on a PC with Intel Core 2 CPU 2.40Ghz and 3G memory. `CVX` [54], a MATLAB package for specifying and solving convex programs, is used to solve all the convex programming problems stated in *AEA-ETRS*. The tolerance parameter ϵ is set to be 0.001.

Example 4.4.1. Consider the following problem from [125]:

$$\begin{aligned} \min \;\; & -x_1^2 + x_1 + 4x_2^2 \\ \text{s.t.} \;\; & x_1^2 + x_2^2 \leq 4, \\ & x_1^2 - 4x_1 + \frac{1}{4}x_2^2 \leq 0. \end{aligned}$$

The initial rectangle set $\mathcal{T}_1 = [0,\ 2] \times [-1.9826,\ 1.9826]$. This rectangle generates the initial ellipsoid $\mathcal{E}_1 = \{x \in \mathbb{R}^2 | (x_1 - 1)^2 + \frac{x_2^2}{1.9826^2} \leq 1\}$. By solving problem (RLT-ETRS), it returns the optimal value -2.5 and the most sensitive point $x^* = (2.4142,\ 0.0000)^T$. By solving (I^c), it returns the approximate solution $\bar{x} = (2.0000,\ 0.0000)^T$, which is the global minimizer. The rectangle $\mathcal{T}_1$ is then split by line $x_2 = 0$ such that $\mathcal{T}_2 = [0,\ 2] \times [0,\ 1.9826]$ and $\mathcal{T}_3 = [0,\ 2] \times [-1.9826,\ 0]$. The upper bound is -2 and the lower bound is -2.5 after the first iteration. Some of the upper and lower bounds generated in the first 40 iterations are shown in Table 4.1.

Table 4.1: The upper and lower bounds of Example 4.4.1.

Iteration	1	5	10	15	30	40
Upper bound	−2.0000	−2.0000	−2.0000	−2.0000	−2.0000	−2.0000
Lower bound	−2.5000	−2.0131	−2.0120	−2.0104	−2.0025	−2.0008

Figure 4.1 depicts the upper and lower bounds returned in each iteration. At the 40th iteration, the algorithm stops as the gap between the upper and lower bounds becomes 0.0008. It outputs an approximate solution $\tilde{x} = (2.0000,\ 0.0000)^T$.

Example 4.4.2. This example also comes from [125]:

$$\begin{aligned} \min \;\; & -x_1^2 + x_1 + x_2^2 \\ \text{s.t.} \;\; & x_1^2 + x_2^2 \leq 4, \\ & (x_1 + x_2)^2 + x_2^2 - 2x_1 \leq 0. \end{aligned}$$

The optimal solution is $x^* = (2,\ 0)^T$ with the optimal value -2. It took 8 iterations for *AEA-ETRS* to terminate. The algorithm provides a lower bound -2.0008, an upper bound -2.0000 and an approximate solution $\tilde{x} = (2.0000,\ 0.0000)^T$. The upper and lower bounds returned

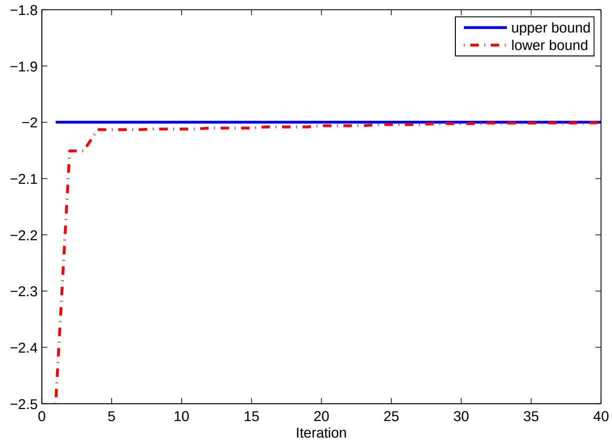

Figure 4.1: The upper and lower bounds of Example 4.4.1.

from each iteration are shown in Table 4.2.

Table 4.2: The upper and lower bound of Example 4.4.2.

Iteration	1	2	3	4	5	6	7	8
UB	−2.0000	−2.0000	−2.0000	−2.0000	−2.0000	−2.0000	−2.0000	−2.0000
LB	−2.3336	−2.1641	−2.1950	−2.1089	−2.0658	−2.0408	−2.0301	−2.0008

Example 4.4.3. Consider the following nonconvex quadratic problem with three convex constraints.

$$\begin{aligned} \min \quad & -x_1^2 + x_1 + 4x_2^2 \\ \text{s.t.} \quad & x_1^2 + x_2^2 \leq 4, \\ & x_1^2 - 4x_1 + \frac{1}{4}x_2^2 \leq 0. \\ & x_1 + x_2 \leq 2 \end{aligned}$$

The optimal solution of this problem is $x^* = (2,\ 0)^T$ with the optimal value -2. Our algorithm terminated in 25 iterations with an upper bound -2.0000 and a lower bound -2.0009. The approximate solution obtained is $\tilde{x} = (2.0000, 0.0000)^T$ which is quite close to the true optimal solution x^*.

4.5 Computational Results

In this section, we report some computational results of the proposed *AEA-ETRS*. The test problems have the following form:

$$\begin{aligned} \min \quad & x^T P^0 x + 2(q^0)^T x \\ \text{s.t.} \quad & x^T P^j x + 2(q^j)^T x + \gamma^j \le 0, \quad j = 1, 2, \ldots, m, \\ & Ex \le d \end{aligned}$$

where the matrices $P^j = O^j D^j (O^j)^T$ for some orthogonal matrix O^j and a diagonal matrix $D^j \in \mathcal{S}^n_{++}$, E is an $l \times n$ matrix and $d \in \mathbb{R}^l$. The parameters in the test problems are randomly generated using the method in [128]. The matrix $O^j = Q_1^j Q_2^j Q_3^j$, where $Q_i^j = I - 2\dfrac{\omega_i \omega_i^T}{\|\omega_i\|^2}$, $i = 1, 2, 3$. The components of vector $\omega_i \in \mathbb{R}^n$ are random numbers in $[-1, 1]$ and I is the n-dimensional identity matrix. The components in D^j are set in the following way: $D^0 = \text{Diag}(D_1^0, ..., D_n^0)$ is a diagonal matrix with $D_i^0 \in [-50, 0]$ for $i = 1, ..., \lfloor \frac{n}{2} \rfloor$ and $D_i^0 \in [0, 50]$ for $i = \lfloor \frac{n}{2} \rfloor + 1, .., n$; $D^j = \text{Diag}(D_1^j, ..., D_n^j)$ is a diagonal matrix with $D_i^j \in [0, 50]$ for $j = 1, ..., m$. The parameters in the linear constraints are set in the following way: $E = (E_{ij})$ with $E_{ij} \in (0, 50)$ and $d = Ee/n$, where e is the vector with all 1s in $\mathbb{R}^n$. Also, we set $q^0 = (q_1^0, ..., q_n^0)^T$ with $q_i^0 \in [-10, 10]$, $q^j = (q_1^j, ..., q_n^j)^T$ with $q_i^j \in [-50, 0]$, and $\gamma^j \in [-5, 0]$ for $j = 1, ..., m$.

In order to demonstrate the validity of *AEA-ETRS*, we used the commercial global optimization package BARON [97] on NEOS server [41] to obtain the optimal value. The termination condition was set as $|s - l| < \epsilon |l| + \epsilon$ with $\epsilon = 10^{-5}$. For $n = 10$, $m = 5$ and $l = 0$, 100 random test problems were generated and Table 4.3 lists part of the results, including the optimal values returned by BARON and *AEA-ETRS*, the number of iterations for *AEA-ETRS* and the CPU time (in seconds) consumed by BARON and *AEA-ETRS*, respectively. Among these 100 problems, 74 problems were solved in one iteration by *AEA-ETRS* and the results coincided with BARON's. For the remaining 26 problems, we selected 10 representative problems to report their results in Table 4.3. It indicates that the proposed *AEA-ETRS* could obtain comparable accurate solutions with BARON in an efficient manner.

In order to show the efficiency of *AEA-ETRS*, we compare our duality gap with the one obtained by Shor's SDP relaxation scheme. To measure the improvement of duality gap obtained by *AEA-ETRS*, the following improvement ratio is adopted[128]:

$$\text{improv. ratio} \triangleq \left(1 - \frac{s - l}{UP - LB}\right) \times 100\%$$

where UP and LB are the upper and lower bounds obtained using Shor's SDP relaxation scheme, s and l are the upper and lower bounds obtained using *AEA-ETRS*. Since the number

Table 4.3: Numerical results for 10 test problems.

Instance	OPT		Iterations	CPU Time (second)	
	BARON	*AEA-ETRS*		BARON	*AEA-ETRS*
1	−1.6641E+4	−1.6641E+4	108	1018.42	322.01
2	−5.2393E+3	−5.2393E+3	127	896.27	400.59
3	−9.3390E+3	−9.3390E+3	117	854.17	347.56
4	−1.4626E+4	−1.4626E+4	99	394.17	286.19
5	−2.3251E+4	−2.3251E+4	43	94.20	134.33
6	−2.5182E+4	−2.5182E+4	507	592.85	1387.94
7	−1.3063E+4	−1.3063E+4	211	900.59	601.71
8	−7.2712E+3	−7.2712E+3	210	779.17	600.41
9	−1.6559E+4	−1.6559E+4	206	671.90	602.15
10	−8.3801E+3	−8.3801E+3	183	2189.81	600.04

of iterations to reach a precise solution may grow fast as the problem size grows, the maximum allowed CPU time was limited to 600 seconds. Table 4.4 summarizes the average improvement ratios, the average number of iterations and the average CPU time for 100 test problems that took more than one iteration to stop for $n = 30, 40, 50$ and $m = l = 5$ or 10. From the results in Table 4.4, we can see that *AEA-ETRS* can close the duality gap resulting from Shor's SDP relaxation scheme very efficiently.

Table 4.4: Numerical results for random generated instances

n	$m = l$	improv. ratio (%)	# iteration	CPU time(second)
30	5	90.16	98.56	600.00
30	10	85.80	91.68	600.00
40	5	94.49	55.58	600.00
40	10	89.43	53.92	600.00
50	5	93.87	30.05	600.00
50	10	91.81	27.62	600.00

4.6 Summary

In this chapter, we have developed a conic reformulation and approximation to a nonconvex quadratic programming problem with several convex quadratic constraints. Because the cone of nonnegative quadratic functions $\mathscr{N}_\Delta$ over the feasible domain Δ of problem (ETRS) and its dual cone $\mathscr{N}_\Delta^*$ are uncomputable, we use the cone of nonnegative quadratic functions $\mathscr{N}_\mathscr{E}$

over an elliptic cover $\mathscr{E}$ of Δ and its dual cone $\mathscr{N}_{\mathscr{E}}^*$ to approximate the cone $\mathscr{N}_{\Delta}$ and $\mathscr{N}_{\Delta}^*$, respectively. The linear conic programming problem (RLT-ETRS) over the cone $\mathscr{N}_{\mathscr{E}}^*$ is actually a semidefinite programming problem. Thus, problem (RLT-ETRS) can be solved efficiently and its optimal value provides a lower bound of problem (ETRSP). In order to obtain a lower bound close to the optimal value of problem (ETRS), an adaptive scheme is adopted to refine the elliptic cover $\mathscr{E}$. The proposed algorithm is shown to be convergent. It is important to point out that our algorithm requires a weaker assumption on the feasible domain Δ than other constraint qualifications ([37, 117, 125]) and the number of convex quadratic constraints can be any positive integer. Hence, our work expands the known results in literature.

There are two main results obtained in this chapter: (i) We use the linear conic programming problem over the cone of nonnegative quadratic functions with new RLT-constraints to approximate the original problem. This may result in better lower bounds than solely applying the SDP or other methods. (ii) We use an adaptive scheme to improve the lower bounds such that the information of the objective function is involved in each iteration to help the proposed algorithm converge to the optimal solution quickly.

However, we have to mention that the arrangement of the ellipsoids $\mathcal{E}_i$ in the elliptic cover $\mathscr{E}$ is critical to our algorithm. The half axes of the ellipsoids used in our algorithm are all parallel to the coordinate axes. This may not be the most efficient way for approximating the original feasible domain. The reason we adopted this arrangement is due to its simplicity in managing and refining the ellipsoid covers. Developing efficient approximation methods is our future research directions.

Although the proposed algorithm is designed to solve a special class of (QCQP), this algorithm can be readily extended to solve other nonconvex quadratic programming problems over some special feasible domains. One such domain is $\Delta' = \Delta_1 \cup \cdots \cup \Delta_k \subseteq \mathbb{R}^n$, where each set $\Delta_i = \{x \in \mathbb{R}^n | g_i^j(x) \leq 0, j = 1, ..., m_i\}$ is bounded and has a nonempty interior with $g_i^j(x)$ being convex for $i = 1, ..., k$ and $j = 1, ..., m_i$. For each of such set Δ_i, we can solve the nonconvex quadratic programming problem over Δ_i using our proposed algorithm with simple modifications. Then, the optimal value of the problem over Δ' can be obtained by comparing the optimal values of the problem over each Δ_i for $i = 1, ..., k$.

Chapter 5

Bounded Quadratically Constrained Quadratic Programming Problems

In this chapter, we study the bounded quadratically constrained quadratic programming (BQCQP) problem. The BQCQP problem is first transformed into a linear conic programming problem, and then approximated by semidefinite programming (SDP) problems over different intervals. In order to improve the lower bounds, polar cuts, generated from the cut-generation problems, and disjunctive cuts are embedded in a branch-and-cut algorithm, which yields a globally ϵ_r-ϵ_z-optimal solution (with respect to feasibility and optimality respectively) in a finite number of iterations. In order to enhance the computational speed, a special branch-and-cut rule is adopted. Numerical examples show that the number of explored nodes can be significantly reduced.

5.1 Introduction

In this chapter, we study the bounded quadratically constrained quadratic programming problem in the following form:

$$\text{(BQCQP)} \qquad \begin{array}{lll} \min & f(x) = x^T P^0 x + 2(q^0)^T x & \\ \text{s.t.} & x^T P^j x + 2(q^j)^T x + \gamma^j \leq 0, & j = 1, 2, \ldots, m, \\ & l \leq x \leq u, & \end{array} \tag{5.1}$$

where l and u are vectors in $\mathbb{R}^n$, $P^j \in \mathbb{R}^{n \times n}$ is a real symmetric matrix, $q^j \in \mathbb{R}^n$ is a vector, $j = 0, \ldots, m$, and $\gamma^j \in \mathbb{R}$ is a scalar, $j = 1, \ldots, m$. Let $\mathcal{F}$ be the feasible domain of problem (BQCQP), i.e.,

$$\mathcal{F} = \left\{ x \in \mathbb{R}^n | l \leq x \leq u,\ x^T P^j x + 2(q^j)^T x + \gamma^j \leq 0,\ j = 1, 2, \ldots, m \right\}.$$

Since the feasible domain $\mathcal{F}$ is bounded and closed, it is compact. However, we do not assume any convexity of the objective function $f(x)$ or the feasible domain $\mathcal{F}$.

Problem (BQCQP) arises from many applications including signal processing [77], statistics [111] and finance [130]. It generalizes many well-known NP-hard optimization problems, such as box constrained quadratic programming, mix 0-1 linear programming, bilinear programming and polynomial programming problems. Hence problem (BQCQP) is NP-hard in general, although some special cases of problem (BQCQP) can be solved in polynomial time [19, 26, 125, 127].

Most global optimization methods for BQCQP are based on the convex relaxations of the original problem embedded in a branch-and-bound (or branch-and-cut) framework where a lower or upper bound is computed by some relaxation schemes, such as linearization [9, 13, 106, 109], second-order cone programming (SOCP) [68] and semidefinite programming (SDP) [49]. A branch-and-bound algorithm based on the outer polyhedral approximations over a rectangle and linear programming relaxations for BQCQP was proposed by Al-Khayyal et al. in [9]. Linderoth [72] and Raber [94] extended this work and developed a branch-and-bound algorithm involving linear programming subproblems based on a simplicial partition. Linear programming relaxations based on reformulation-and-linearization techniques (RLT) were developed in [104, 106]. Audet et al. [13] extended the use of RLT in solving BQCQP by including different classes of linearizations. Kim and Kojima [68] applied the lift-and-project idea of RLT to create a second-order cone programming relaxation for BQCQP. Notice that the relaxation problems in all these works are linear programming (LP) or second-order cone programming (SOCP) problems.

SDP relaxation is another attractive approach for solving problem (BQCQP) owing to its capability of finding good bounds and approximation solutions. By lifting a vector $x \in \mathbb{R}^n$ to a symmetric positive semidefinite matrix $X = xx^T$, which can be represented by the linear matrix inequality $X \succcurlyeq xx^T$ ("$\succcurlyeq$" means $X - xx^T$ is positive semidefinite) together with linear inequalities $x_i^2 \geq X_{ii}$ for $i = 1, ..., n$, we see that problem (BQCQP) is equivalent to the following linear conic problem (ref. Theorem 3.5 in [35]):

$$\text{(CP)} \qquad \begin{array}{lll} \min & F(x,X) = P^0 \bullet X + (q^0)^T x & \\ \text{s.t.} & P^j \bullet X + (q^j)^T x + \gamma^j \leq 0, & j = 1, \ldots, m, \\ & x_i^2 \geq X_{ii}, & i = 1, \ldots, n, \\ & X \succcurlyeq xx^T, \; l \leq x \leq u. & \end{array} \qquad (5.2)$$

Note that $X \succcurlyeq xx^T$ is a convex constraint because it is equivalent to $\begin{bmatrix} 1 & x^T \\ x & X \end{bmatrix} \succcurlyeq 0$ according to Schur complementary theorem [20]. However, $x_i^2 \geq X_{ii}$ is nonconvex for $i = 1, \ldots, n$. Thus solving problem (CP) directly is likely to be NP-hard [78]. Instead, various convex relaxations for problem (CP) were proposed. For example, the Shor's relaxation scheme [110] eliminates the nonconvex constraints $x_i^2 \geq X_{ii},\ i = 1, ..., n$, in problem (CP), resulting in a basic SDP relaxation as follows.

$$\text{(Shor)} \qquad \begin{array}{ll} \min & P^0 \bullet X + (q^0)^T x \\ \text{s.t.} & P^j \bullet X + (q^j)^T x + \gamma^j \leq 0, \quad j = 1, ..., m, \\ & X \succcurlyeq xx^T,\ l \leq x \leq u. \end{array} \tag{5.3}$$

Anstreicher derived another relaxation in [11] by applying RLT, which linearizes the bilinear term $x_i x_j$ by X_{ij} and applies some valid bounds on $x_i x_j$ to X_{ij}. Specifically, the resulting SDP relaxation is as follows.

$$\text{(SDP-RLT)} \qquad \begin{array}{ll} \min & P^0 \bullet X + (q^0)^T x \\ \text{s.t.} & P^j \bullet X + (q^j)^T x + \gamma^j \leq 0,\ j = 1, ..., m, \\ & X \succcurlyeq xx^T,\ l \leq x \leq u,\ X_{ij}^- \leq X_{ij} \leq X_{ij}^+,\ i, j = 1, ..., n, \end{array} \tag{5.4}$$

where $X_{ij}^- = \max\{u_i x_j + u_j x_i - u_i u_j, l_i x_j + l_j x_i - l_i l_j\}$ and $X_{ij}^+ = \min\{l_i x_j + u_j x_i - l_i u_j, u_i x_j + l_j x_i - u_i l_j\}$. Further eliminating the semidefinite constraint $X \succcurlyeq xx^T$ in problem (SDP-RLT), we have the following linear relaxation (ref. [106]):

$$\text{(LP-RLT)} \qquad \begin{array}{ll} \min & P^0 \bullet X + (q^0)^T x \\ \text{s.t.} & P^j \bullet X + (q^j)^T x + \gamma^j \leq 0,\ j = 1, ..., m, \\ & l \leq x \leq u,\ X_{ij}^- \leq X_{ij} \leq X_{ij}^+,\ i, j = 1, ..., n. \end{array} \tag{5.5}$$

Denote the feasible domains of problem (CP), (Shor), (SDP-RLT) and (LP-RLT) by $\mathcal{F}_{\text{CP}}$, $\mathcal{F}_{\text{Shor}}$, $\mathcal{F}_{\text{SDPRLT}}$ and $\mathcal{F}_{\text{LPRLT}}$, respectively. Note that, $\mathcal{F}_{\text{Shor}}$ and $\mathcal{F}_{\text{SDPRLT}}$ are outer-approximations of $\mathcal{F}_{\text{CP}}$, and $\mathcal{F}_{\text{CP}} \subseteq \mathcal{F}_{\text{SDPRLT}} \subseteq \mathcal{F}_{\text{Shor}}$. The set $\mathcal{F}_{\text{LPRLT}}$ is a polyhedral outer-approximation of $\mathcal{F}_{\text{CP}}$ and problem (LP-RLT) is linear programming, thus can be solved more efficiently than problems (Shor) and (SDP-RLT) in general.

In this chapter, a new branch-and-cut algorithm based on the estimation of quadratic terms x_i^2, $i = 1, \ldots, n$, is developed. The similar technique has appeared in [13], but is different from ours in two respects. First, only the quadratic terms x_i^2, $i = 1, \ldots, n$, are needed to approximate are needed to improve and approximate in different intervals in our work, while other researchers estimate all bilinear terms $x_i x_j$, $i, j = 1, \ldots, n$ (ref. [11, 13, 72]). Second, we propose new linear and quadratic polar cuts in the cutting step. All these cuts are valid in the subtrees rooted at the nodes where they were generated. The linear polar cuts in [98] and [99] are generated by solving linear cut-generation problems. However, the polar cuts in this study are generated by solving SDP problems, which include the cut-generation problems in [99] as special cases.

Consequently, the new cuts may improve the bounds more efficiently.

This chapter is organized as follows. Section 5.2 introduces the range reduction technique to tighten the bounds on the variables. In Section 5.3, three different cuts, including the linear polar cut, quadratic polar cut and disjunctive cut, are derived. A branch-and-cut algorithm, which converges in a finite number of iterations within a given tolerance, is proposed in Section 5.4. Numerical examples are reported in Section 5.5. Conclusions are given in Section 5.6.

5.2 Range Reduction Strategy

A tight bound on variables $\{x_i\}$ will reduce the search space to find the optimal solution more quickly. Due to the nonconvex nature of the constraints in problem (BQCQP), obtaining tight bounds on the variables $\{x_i\}$ is a nontrivial job. The range reduction strategy we used is the one in [105]. Let x_i^-, x_i^+ be the current bounds on variable x_i such that $0 \le x_i^- \le x_i \le x_i^+$, and X_{ii}^-, X_{ii}^+ be the bounds on X_{ii} such that $0 \le X_{ii}^- \le X_{ii} \le X_{ii}^+$ obtained by replacing problem (LP-RLT)'s objective function by $\pm x_i$ and then by $\pm X_{ii}$, respectively. We then update the bound l_i and u_i according to $l_i = \max\{x_i^-,\ \sqrt{X_{ii}^-}\}$ and $u_i = \min\{x_i^+,\ \sqrt{X_{ii}^+}\}$, respectively. If $l_i \le u_i$, then l_i and u_i are valid bounds for variable x_i over the feasible domain of problem (BQCQP). Otherwise, $l_i > u_i$ and the feasible domain of problem (BQCQP) is empty.

After updating the bound for one variable x_i, the process can be reiterated for other variables $x_j (j \neq i)$ to tighten the bound. For demonstration, consider the following example from [13]:

Example 5.2.1. Let the feasible domain $\mathcal{F}$ be defined by two constraints

$$x_1 + x_1^2 \le 6 \quad \text{and} \quad x_1 \ge 1.$$

It can be verified that the feasible domain is $[1,\ 2]$. The range reduction strategy first solves the following two problems

$$\begin{aligned} x_1^- = \min \quad & x_1 \\ \text{s.t.} \quad & x_1 + X_{11} \le 6,\ x_1 \ge 1, \\ & X_{11} \ge 2x_1 - 1, \end{aligned} \tag{5.6}$$

and

$$\begin{aligned} X_{11}^- = \min \quad & X_{11} \\ \text{s.t.} \quad & x_1 + X_{11} \le 6,\ x_1 \ge 1, \\ & X_{11} \ge 2x_1 - 1, \end{aligned} \tag{5.7}$$

and finds lower bounds on x_1 and X_{11} are $x_1^- = 1$ and $X_{11}^- = 1$, respectively. The lower bound

l_1 on x_1 is then updated to $l_1 = \max\left\{x_1^-, \sqrt{X_{11}^-}\right\} = 1$. Next, the range reduction strategy solves the following two problems

$$\begin{aligned} x_1^+ = \max \quad & x_1 \\ \text{s.t.} \quad & x_1 + X_{11} \leq 6, \;\; x_1 \geq 1, \\ & X_{11} \geq 2x_1 - 1, \end{aligned} \tag{5.8}$$

and

$$\begin{aligned} X_{11}^+ = \max \quad & X_{11} \\ \text{s.t.} \quad & x_1 + X_{11} \leq 6, \;\; x_1 \geq 1, \\ & X_{11} \geq 2x_1 - 1, \end{aligned} \tag{5.9}$$

and finds the upper bounds on x_1 and X_{11} are $x_1^+ = \frac{7}{3}$ and $X_{11}^+ = 5$, respectively. The upper bound u_1 on x_1 is then updated to $u_1 = \min\left\{x_1^+, \sqrt{X_{11}^+}\right\} = \sqrt{5}$. The new bound on x_1 then generate new RLT-contraints:

$$X_{11} - 2\sqrt{5}x_1 + 5 \geq 0 \text{ and } -X_{11} + (1+\sqrt{5})x_1 - \sqrt{5} \geq 0.$$

By adding the new RLT-constraints along with the new bound on x_1 to problems (5.6)–(5.9) and resolving them, the range reduction strategy then finds the new bound on x_1 is $\left[1, \; \frac{11}{2\sqrt{5}+1}\right]$. Repeating this process converges to the feasible interval of x_1, i.e., $[1, \; 2]$.

In Section 5.4, we will present an algorithm whose pre-processing phase adopts this range reduction strategy until the improvement on the bounds becomes negligible.

5.3 Cuts Generation

In this section, three different cuts for the new branch-and-cut algorithm are developed. We first derive the linear and quadratic polar cuts, which are available when the optimal values of corresponding cut-generation problems are negative. Then we introduce the disjunctive cuts, which are always available.

5.3.1 Generalized Linear and Quadratic Polar Cuts

Assume the current solution of problem (SDP-RLT) is (x^*, X^*) . If $X^* = x^*(x^*)^T$, then x^* is an optimal solution of problem (CP). Otherwise, a valid (linear or nonlinear) inequality is expected to cut off the current solution (x^*, X^*) such that a tighter lower bound could be obtained in the next iteration. Such valid inequality is called a *cut* for problem (CP). There are various types of cuts, such as intersection cuts [15], disjunctive cuts [16] and polar cuts [99], in the literature. We generalize the so-called *linear polar cuts* by extending the results of [99]. It turns out that the extension is nontrivial and the following lemma is needed.

Lemma 5.3.1. *For a given positive semidefinite matrix $A \in \mathcal{S}^n_+$ and a vector $\lambda = (\lambda_1, \ldots, \lambda_K)^T \in \mathbb{R}^K$ with $K \in \mathbb{N}$, $\sum_{k=1}^K \lambda_k = 1$ and $\lambda_k \geq 0$ for $k = 1, \ldots, K$. The following inequality holds*

$$A \bullet (\sum_{k=1}^{K} \lambda_k v^k)(\sum_{l=1}^{K} \lambda_l v^l)^T \leq \sum_{k=1}^{K} \lambda_k A \bullet (v^k)(v^k)^T \tag{5.10}$$

for any set of $\{v^k \in \mathbb{R}^n,\ k = 1, \ldots, K\}$.

Proof. Let $a = A \bullet (\sum_{k=1}^K \lambda_k v^k)(\sum_{l=1}^K \lambda_l v^l)^T - \sum_{k=1}^K \lambda_k A \bullet (v^k)(v^k)^T$, we need to prove $a \leq 0$. Note that

$$\begin{aligned}
a &= \sum_{k=1}^{K} \lambda_k (v^k)^T A(\sum_{l=1}^{K} \lambda_l v^l - v^k) \\
&= \sum_{k=1}^{K} \sum_{l=1}^{K} \lambda_k \lambda_l (v^k)^T A(v^l - v^k) \\
&= \sum_{k=1}^{K} \sum_{l=1}^{K} \lambda_k \lambda_l (v^k - v^l)^T A(v^l - v^k) + \sum_{k=1}^{K} \sum_{l=1}^{K} \lambda_k \lambda_l (v^l)^T A(v^l - v^k) \\
&= -\sum_{k=1}^{K} \sum_{l=1}^{K} \lambda_k \lambda_l (v^l - v^k)^T A(v^l - v^k) - a \\
&\leq -a
\end{aligned}$$

where the last inequality holds because A is positive semidefinite and $\lambda_k \geq 0$ for $k = 1, ..., K$. Hence $a \leq 0$ and inequality (5.10) holds. □

For a given set $\mathcal{S} \subseteq \{1, 2, ..., n\}$, denote $|\mathcal{S}|$ to be its cardinality. Let $x_{\mathcal{S}} = \langle x_k \rangle_{k \in \mathcal{S}}$ be the sub-vector of x having components indexed by $k \in \mathcal{S}$, and $X_{\mathcal{S}} = \langle X_{ij} \rangle_{i,\ j \in \mathcal{S}}$ be the sub-matrix of X having component indexed by $i,\ j \in \mathcal{S}$. We have the following theorem generalizing the linear polar cuts.

Theorem 5.3.1. *Let $\mathcal{S} \subseteq \{1, 2, ..., n\}$, $\mathcal{P} \subseteq \mathbb{R}^{|\mathcal{S}|}$ be a polyhedral outer-approximation of $\Omega_{\mathcal{S}} = \{x_{\mathcal{S}} = \langle x_k \rangle_{k \in \mathcal{S}} \big| \exists \langle x_k \rangle_{k \notin \mathcal{S}}$ and X such that $(x, X) \in \mathcal{F}_{\mathrm{CP}}\}$, which is the projection of $\mathcal{F}_{\mathrm{CP}}$ to the space of $\langle x_k \rangle_{k \in \mathcal{S}}$ variables, and $V = \{v^t | t = 1, \ldots, K\}$, where $K \in \mathbb{N}$ and $v^t \in \mathbb{R}^{|\mathcal{S}|}$ for $t = 1, ...K$, be the set of vertices of $\mathcal{P}$. Then, a given (x^*, X^*) is a feasible solution to problem* (CP) *only if the optimal objective value of the following linear cut-generation problem is nonnegative.*

$$
\text{(LPolar)} \qquad \begin{array}{ll} \min & \alpha^T x^*_{\mathcal{S}} + B \bullet X^*_{\mathcal{S}} - \gamma \\ \text{s.t.} & \alpha^T v^t + B \bullet (v^t (v^t)^T) - \gamma \geq 0, \ t = 1, \dots, K, \\ & \alpha = \alpha^+ - \alpha^-, \\ & \alpha^+ \geq 0, \ \alpha^- \geq 0, \ -B \succcurlyeq 0, \\ & e^T(\alpha^+ + \alpha^-) - \text{tr}(B) = 1. \end{array} \tag{5.11}
$$

Moreover, if $(\alpha, B, \gamma, \alpha^+, \alpha^-)$ *is a feasible solution of problem* (LPolar) *having a negative objective value, then* $\alpha^T x_{\mathcal{S}} + B \bullet X_{\mathcal{S}} - \gamma \geq 0$ *cuts off* (x^*, X^*) *from the feasible domain of problem* (CP).

Proof. Suppose (x, X) is a feasible solution to problem (CP) and $(\alpha, B, \gamma, \alpha^+, \alpha^-)$ is a feasible solution to problem (LPolar). Since $(x, X) \in \mathcal{F}_{\text{CP}}$, we have $x_{\mathcal{S}} = \langle x_k \rangle_{k \in \mathcal{S}} \in \Omega_{\mathcal{S}}$, $X_{\mathcal{S}} = x_{\mathcal{S}} x_{\mathcal{S}}^T$ and there exist some $\lambda_t \geq 0$ for $t = 1, \dots, K$ such that $\sum_{t=1}^K \lambda_t = 1$ and $\sum_{t=1}^K \lambda_t v^t = x_{\mathcal{S}}$. Consequently,

$$
\begin{aligned}
& \alpha^T x_{\mathcal{S}} + B \bullet X_{\mathcal{S}} - \gamma \\
= & \left(\sum_{t=1}^K \alpha^T \lambda_t v^t + B \bullet (\sum_{t=1}^K \lambda_t v^t)(\sum_{t=1}^K \lambda_t v^t)^T \right) - \gamma \\
\geq & \sum_{t=1}^K \lambda_t \left(\alpha^T v^t + B \bullet (v^t)(v^t)^T \right) - \gamma \quad \text{(by Lemma 5.3.1 and the positive semidefiniteness of } -B) \\
\geq & \ 0.
\end{aligned}
$$

Therefore, $\alpha^T x_{\mathcal{S}} + B \bullet X_{\mathcal{S}} - \gamma \geq 0$ is a valid inequality for problem (CP). When the optimal objective value of problem (LPolar) is negative, (x^*, X^*) violates the inequality $\alpha^T x_{\mathcal{S}} + B \bullet X_{\mathcal{S}} - \gamma \geq 0$ and is cut off from the feasible domain of problem (CP). □

Remark 8. When matrix B is restricted to be diagonal, problem (LPolar) is the same as the one in [99]. Hence, our result is more general and has a better chance to improve the lower bound more efficiently. Thus our cut is called a "generalized linear polar cut." The constraint of $e^T(\alpha^+ + \alpha^-) - \text{tr}(B) = 1$ is a normalization constraint that ensures the boundedness of problem (LPolar).

Problem (LPolar) is an SDP problem, thus can be solved in polynomial time [8]. Furthermore, we can extend the results in Theorem 5.3.1 to a quadratic case. The next theorem shows the way to obtain a quadratic polar cuts.

Theorem 5.3.2. *Let $\mathcal{S} \subseteq \{1,2,...,n\}$, $\mathcal{P} \subseteq \mathbb{R}^n$ be a polyhedral outer-approximation of $\Omega_{\mathcal{S}}$ and $V = \{v^t | t = 1,\ldots,K\}$ be the set of vertices of $\mathcal{P}$, where $K \in \mathbb{N}$ and $v^t \in \mathbb{R}^{|\mathcal{S}|}$ for $t = 1,...K$. Then, a given (x^*, X^*) is a feasible solution to problem* (CP) *only if the optimal objective value of the following quadratic cut-generation problem is nonnegative.*

$$
\text{(QPolar)} \qquad \begin{array}{ll} \min & (x^*_{\mathcal{S}})^T A x^*_{\mathcal{S}} + B \bullet X^*_{\mathcal{S}} - \gamma \\ \text{s.t.} & (A+B) \bullet v^t (v^t)^T - \gamma \geq 0, \ t = 1,\ldots,K, \\ & -(A+B) \succcurlyeq 0, \\ & -\text{tr}(A+B) = 1. \end{array} \tag{5.12}
$$

Moreover, if (A, B, γ) is a feasible solution of problem (QPolar) *having a negative objective value , then $x_{\mathcal{S}}^T A x_{\mathcal{S}} + B \bullet X_{\mathcal{S}} - \gamma \geq 0$ cuts off (x^*, X^*) from the feasible domain of problem* (CP)*.*

Proof. The proof is similar to the one in Theorem 5.3.1 by noting that the second inequality in the proof of Theorem 5.3.1 holds due to the fact that matrix $-(A+B)$ is positive semidefinite. □

Remark 9. The valid cut generated by problem (QPolar) is called a "generalized quadratic polar cut," which may not be tractable because the matrix $-A$ could be nonpositive semidefinite. Therefore, we add an extra constraint $-A \succcurlyeq 0$ for the computational experiments in Section 5.5. It is easy to see that Theorem 5.3.2 still holds, although the quality of the generalized quadratic polar cut might be compromised. The constraint of $-\text{tr}(A+B) = 1$ is a normalization constraint that ensures the boundedness of problem (QPolar).

The vertices of the polyhedral outer-approximation $\mathcal{P}$ are required in order to generate the polar cuts in Theorems 5.3.1 and 5.3.2. The homotopy procedure [83] can find a tight polyhedral outer-approximation for any tractable convex relaxation of $\mathcal{F}_{\text{CP}}$ by solving a family of parametric linear programs. Sometimes, this procedure could be time-consuming and we may consider solving the following linear programming problem:

$$
\text{(AP)} \qquad \begin{array}{ll} \max & \sum_{k \in \mathcal{S}} \theta_k x_k \\ \text{s.t.} & P^j \bullet X + (q^j)^T x + \gamma^j \leq 0, \ j = 1,...,m, \\ & l \leq x \leq u, \ X^-_{ij} \leq X_{ij} \leq X^+_{ij}, \ i,j = 1,...,n. \end{array} \tag{5.13}
$$

with different combinations of θ_k to obtain a rough polyhedral outer-approximation. The detailed procedure is described as follows. For some fixed θ_k, $k = 1,\ldots,|\mathcal{S}|$, assume the optimal solution of problem (AP) is $(\bar{x}, \bar{X})$. Then $\sum_{k\in\mathcal{S}} \theta_k x_k \leq \sum_{k\in\mathcal{S}} \theta_k \bar{x}_k$ is a valid inequality for the domain $\Omega_{\mathcal{S}}$ and $\sum_{k\in\mathcal{S}} \theta_k x_k = \sum_{k\in\mathcal{S}} \theta_k \bar{x}_k$ is a facet of the polyhedral outer-approximation $\mathcal{P}$. All these valid inequalities corresponding to different θ along with the bound on x defines the polyhedron $\mathcal{P}$, whose vertices are then used in problems (LPolar) and (QPolar) to generate polar cuts. In our implementation, we use all subsets $\mathcal{S} \subseteq \{1, \ldots, n\}$ with $|\mathcal{S}| = 2$. For each $\mathcal{S} = \{i, j\}$, we compute the facets of the polyhedron $\mathcal{P}$ by solving problem (AP) with $\theta = (u_j - l_j, -u_i + l_i)$ $, \theta = (-u_j + l_j, u_i - l_i), \theta = (u_j - l_j, u_i - l_i)$ and $\theta = (-u_j + l_j, -u_i + l_i)$ sequentially. Vertices of $\mathcal{P}$ are then enumerated by the primal-dual method in [30]. For demonstration, consider the following example from [129]:

Example 5.3.1. Let the domain $\Omega_{\mathcal{S}}$, $\mathcal{S} = \{1, 2\}$, be defined by the following inequalities:

$$\begin{aligned} &2x_1^2 + 4x_1x_2 + 2x_2^2 + 8x_1 + 6x_2 - 9 \leq 0, \\ &-5x_1^2 - 8x_1x_2 - 5x_2^2 - 4x_1 + 4x_2 + 4 \leq 0, \\ &x_1 + 2x_2 \leq 2,\ 0 \leq x_1,\ x_2 \leq 1. \end{aligned} \tag{5.14}$$

The nonconvex domain $\Omega_{\mathcal{S}}$ is shown as the shaded area in Figure 5.1. The range reduction strategy tightens the bounds on x_1 and x_2 to $[0,\ 0.9155]$ and $[0,\ 0.9333]$, respectively. This bounded box is depicted by the dashed lines in Figure 5.1. By sequentially solving problem (AP) with $\theta = (0.9333, -0.9155)$, $\theta = (-0.9333, 0.9155)$, $\theta = (0.9333, 0.9155)$ and $\theta = (-0.9333, -0.9155)$ in turn, we obtain four facets (see the dotted lines in Figure 5.1) of the polyhedral outer-approximation $\mathcal{P}$, whose boundary is the solid lines in Figure 5.1. Combining with the bounds on x_1 and x_2, the vertices of polyhedron $\mathcal{P}$ are $A = (0.4664, 0)$, $B = (0, 0.4754)$, $C = (0, 0.6314)$, $D = (0.1962, 0.8314)$, $E = (0.9155, 0.0981)$ and $F = (0.9155, 0)$.

5.3.2 Disjunctive Cuts

Notice that the optimal objective values of problems (LPolar) and (QPolar) may be nonnegative. In this case, the linear and quadratic polar cuts are not available. This motivates us to find other valid inequalities to cut off the current solution (x^*, X^*) in order to improve the approximation of $\mathcal{F}_{\text{CP}}$. The disjunctive cuts (ref. [16]) can fulfill this purpose.

Assume the current solution (x^*, X^*) satisfies $l_i \leq x_i^* \leq u_i$ for some $i \in \{1, \ldots, n\}$. The disjunctive inequality associated with variable x_i^* is

$$\begin{bmatrix} l_i \leq x_i \leq v_i \\ X_{ii} \leq (l_i + v_i)x_i - l_i v_i \end{bmatrix} \bigvee \begin{bmatrix} v_i \leq x_i \leq u_i \\ X_{ii} \leq (u_i + v_i)x_i - u_i v_i \end{bmatrix} \tag{5.15}$$

where $v_i \in (l_i,\ u_i)$ and "$\vee$" denotes the logical "or", that is x either belongs to the segment of "$l_i \leq x_i \leq v_i,\ X_{ii} \leq (l_i + v_i)x_i - l_i v_i$" or the segment of "$v_i \leq x_i \leq u_i,\ X_{ii} \leq (u_i + v_i)x_i - u_i v_i$". When the disjunctive inequality cuts off (x^*, X^*), we call it a *disjunctive cut*. Obviously, the disjunctive inequality (5.15) cuts off (x^*, X^*) when $v_i = x_i^*$ and $(x_i^*)^2 < X_{ii}^*$. See Figure 5.2 for illustration. We remark that a disjunctive inequality does not increase the complexity because

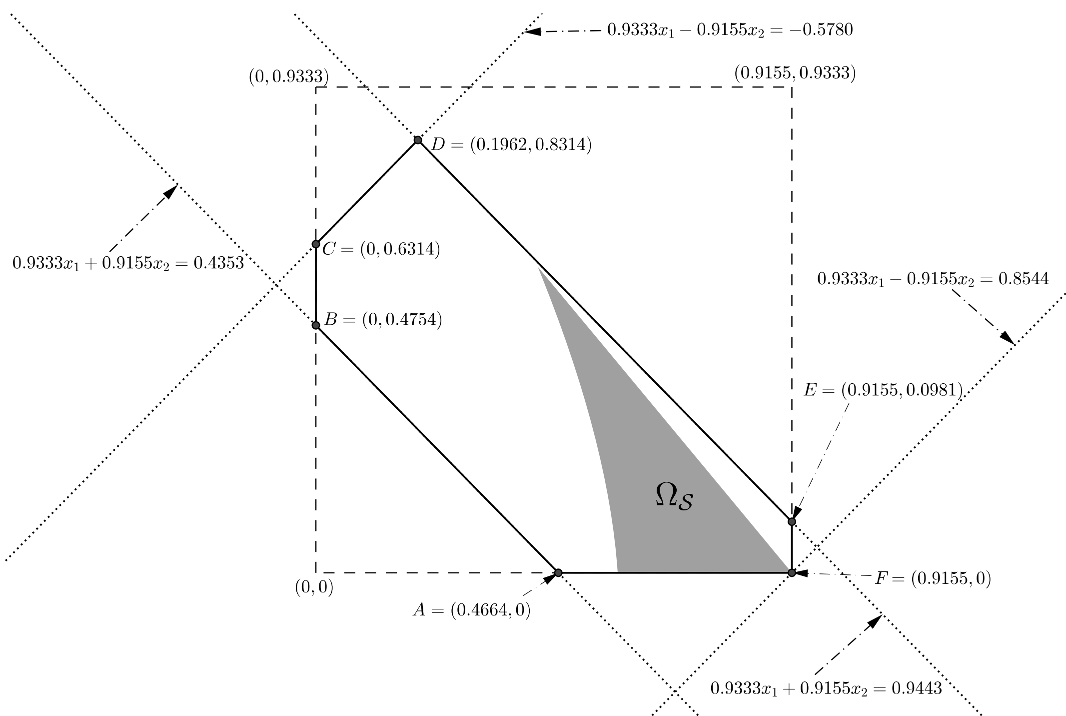

Figure 5.1: Graphic description for Example 5.3.1
The shaded area is the domain $\Omega_{\mathcal{S}}$ described by inequalities in (5.14), the dashed lines are the bounds obtained by range reduction strategy, the dotted lines are the facets obtained by solving problem (AP) with different θ and the solid lines constitute the boundary of the polyhedral outer-approximation $\mathcal{P}$.

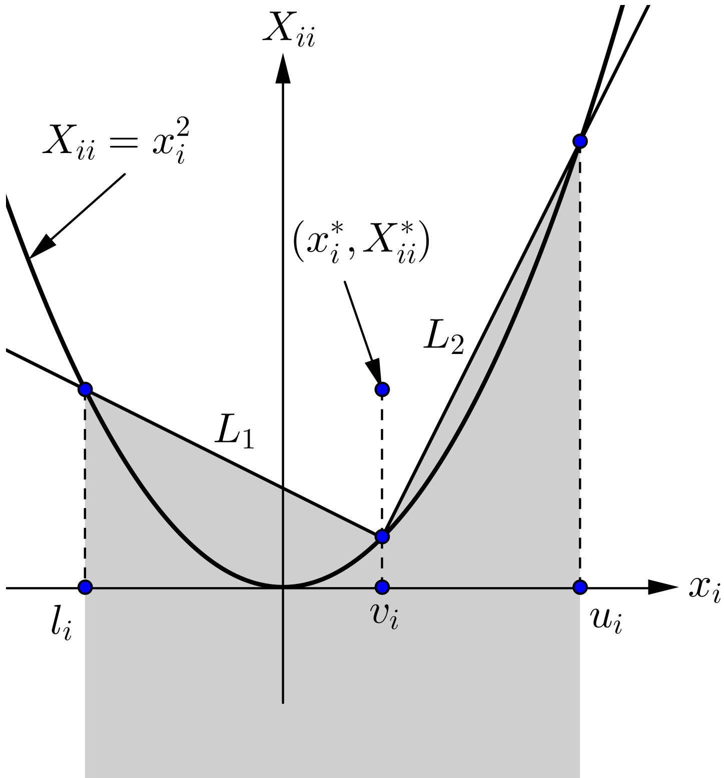

Figure 5.2: A disjunctive cut (5.15) with $v_i = x_i^*$
L_1 is the straight line $X_{ii} = (l_i + x_i^*)x - l_i x_i^*$ and L_2 is the straight line $X_{ii} = (u_i + x_i^*)x - u_i x_i^*$. The shaded area is the domain of disjunctive inequality. Clearly, if $l_i \leq x_i \leq x_i^*$, then (x_i, X_{ii}) must be under line L_1. If $x_i^* \leq x_i \leq u_i$, then (x_i, X_{ii}) must be under line L_2. Since point (x_i^*, X_{ii}^*) is above both L_1 and L_2, it is cut off by this disjunctive inequality.

we can solve the relaxation problems over each segment and choose the best one as the optimal solution.

5.4 A Branch-and-Cut Algorithm

In this section, we utilize the cuts described in Section 5.3 to construct an algorithm for solving problem (CP). We seek an approximation solution in terms of feasibility and optimality.

Definition 7. For a feasibility tolerance parameter $\epsilon_r > 0$, a solution $(x^*, X^*) \in \mathcal{F}_{\mathrm{SDPRLT}}$ is said to be ϵ_r-feasible to problem (CP) if $|X_{ii}^* - (x_i^*)^2| < \epsilon_r$ for $i = 1, \ldots, n$. Moreover, let z^* be the minimal value of $F(x, X)$ over all the ϵ_r-feasible solutions of problem (CP). For an objective tolerance parameter $\epsilon_z > 0$, a solution (x^*, X^*) is said to be ϵ_r-ϵ_z-optimal if it is ϵ_r-feasible and

$|z^* - F(x^*, X^*)| < \epsilon_z$.

The following concepts will be used throughout this section: The branch-and-cut algorithm generates a branching *tree*. The initial node of the tree is called the *root*. When a node is processed and needs further refinement, the branching step creates two new nodes called *children* of the *parent*. The *incumbent solution* refers to the best solution currently found by the algorithm. Its objective value is the *incumbent value* which is set to be $+\infty$ at the beginning of the algorithm.

For the proposed branch-and-cut algorithm, we first describe a pre-processing phase to obtain tight bounds on the variables. Then, we introduce a branching strategy that selects at each node a branching variable and a branching value. Following is a cutting step that refines the approximation of $\mathcal{F}_{\text{CP}}$. Finally, we prove the proposed algorithm finds an approximate solution within any given precision in a finite number of iterations.

5.4.1 Pre-processing phase

At the root node, the initial linear relaxation problem (LP-RLT) is created. Next, bounds are evaluated for each variable x_i, $i = 1, ..., n$, through the iterative process described in Section 5.2. With the new bounds for each variable, the range reduction strategy can be applied again. The pre-processing phase then consists of iteration of this strategy until the improvement of the bounds for each variable is less than a given tolerance ϵ_r.

After the pre-processing phase at the root node, further refinement of the approximation requires generating the polar or disjunctive cuts described in Section 5.3. The algorithm then moves on to a branching step.

5.4.2 Branching step

Assume the solution (x^*, X^*) is obtained by solving problem (SDP-RLT) corresponding to current node. If $(x_i^*)^2 = X_{ii}^*$ for $i = 1, \ldots, n$, then (x^*, X^*) is feasible to problem (CP). The corresponding node is solved and no further branch is needed. Otherwise, there exists $i \in \{1, \ldots, n\}$ such that $(x_i^*)^2 < X_{ii}^*$. Denote all such i by a set $\mathcal{I}$, that is, $\mathcal{I} = \{i \in \{1, ..., n\} | (x_i^*)^2 < X_{ii}^*\} \neq \emptyset$. The purpose of branching is to tighten the current approximation of the nonconvex constraint $x_i^2 = X_{ii}$, $i = 1, \ldots, n$. Therefore, the component of x that violates the constraints $x_i^2 = X_{ii}$ most is selected and the estimation of x_i^2 is further improved in the next iteration. We call the selected component as the *branching variable*. There may exist multiple candidates for the branching variable. In this case, the candidate with the smallest index is chosen. In other words, we choose $i^* = \min\{j \in \mathcal{I} | X_{jj}^* - (x_j^*)^2 = \max_{i \in \mathcal{I}}\{X_{ii}^* - (x_i^*)^2\}\}$ and divide the interval $[l_{i^*},\ u_{i^*}]$ into two parts. Breaking this interval at which point depends on the cuts we will choose in the next step. We point out that the branch-and-cut rule in our algorithm is simpler than the one in [13] because the cardinality of $\mathcal{I}$ is much smaller due to the fact that we do not need to consider the bilinear terms $x_i x_j$ for $i \neq j$, $i, j \in \{1, \ldots, n\}$.

5.4.3 Cutting step

As described in Sections 5.3.1 and 5.3.2, there are three types of valid cuts: linear polar cuts, quadratic polar cuts and disjunctive cuts. The polar cuts are available only if the optimal value of problem (LPolar) or (QPolar) is negative. Therefore, the existence of polar cuts depends on individual cases. For the disjunctive cuts, the parameter v_i in (5.15) decides its shape and quality. The criteria for choosing v_i is to minimize the infeasible domain $\mathcal{F}_{\text{SDPRLT}} \backslash \mathcal{F}_{\text{CP}}$, which is the intersection of the disjunctive inequality (5.15) and the paraboloid $x_i^2 \leq X_{ii}$. The following lemma shows that v_i should be the midpoint of the interval $[l_i, u_i]$ according to this rule.

Lemma 5.4.1. *The area of the intersection of the disjunctive valid inequality* (5.15) *and the paraboloid $x_i^2 \leq X_{ii}$ is minimized when $v_i = \frac{l_i+u_i}{2}$.*

Proof. For any given $v_i \in [l_i,\ u_i]$, the area between (5.15) and the paraboloid $x_i^2 \leq X_{ii}$ is $\frac{1}{2}(u_i - l_i)v_i^2 + \frac{1}{2}(l_i^2 - u_i^2)v_i + \frac{1}{6}(u_i^3 - l_i^3)$, whose minimum is achieved at $v_i = \frac{u_i+l_i}{2}$. □

From Lemma 5.4.1, if the interval $[l_i\ u_i]$ is bisected, then the infeasible area is minimized, which in turn indicates that the disjunctive inequality (5.15) with $v_i = \frac{l_i+u_i}{2}$ cuts deepest. Therefore, dividing the interval at the midpoint is preferred if possible. However, there may exist the case that neither the polar cuts nor the disjunctive inequality with $v_i = \frac{l_i+u_i}{2}$ can cut off the current solution (x_i^*, X_{ii}^*). In this case, we have to use the disjunctive cut with $v_i = x_i^*$ in (5.15). In any of the above case, v_i is called *branching value* for the branching variable x_i.

In summary, our cutting procedure involves the following two steps:

Step 1. Solve problems (LPolar) and (QPolar) for all $\mathcal{S} \subseteq \{1, \ldots, n\}$ with $|\mathcal{S}| = 2$, respectively. If any linear or quadratic polar cut is available, then bisect $[l_i,\ u_i]$ at the middle point $\frac{l_i+u_i}{2}$ into two new intervals $[l_i,\ \frac{l_i+u_i}{2}]$ and $[\frac{l_i+u_i}{2},\ u_i]$ and introduce the polar cut into the relaxation problems corresponding to both children. Otherwise, $[l_i,\ u_i]$ is divided at x_i^* into two new intervals $[l_i,\ x_i^*]$ and $[x_i^*,\ u_i]$.

Step 2. Generate a disjunctive cut based on the two new intervals. If $[l_i,\ u_i]$ is bisected at $\frac{l_i+u_i}{2}$, then generate a disjunctive cut according to (5.15) by setting $v_i = \frac{l_i+u_i}{2}$ and introduce it to the relaxation problems corresponding to both children. Otherwise, generate the disjunctive cut according to (5.15) by setting $v_i = x_i^*$ and introduce it to the relaxation problems corresponding to both children.

From the above description, we can see that our branching value is adaptive. Other adaptive rules for the selection of branching value can be found in [13, 104].

5.4.4 The proposed algorithm

We are now ready to construct our branch-and-cut algorithm. The method can be divided into two main steps. The branching step is used to select the variable to be refined. The cutting step is used to generate a better outer-approximation of the original feasible domain. At each node, one of the two outcomes is possible: either the node is discarded (when infeasible, solved or fathomed), or it is split into two new nodes (children). The nodes are stored in a list and recursively processed in a best-first manner (preference with respect to the optimal objective value of the relaxation). For clarity, only the main ideas of the algorithms are presented. Details were presented in previous three subsections.

Branch-and-Cut Algorithm for BQCQP (BCA-BQCQP)

Pre-processing Phase

Using the range reduction strategy to tighten the bounds of each variable.

Enumeration Phase

The list $\mathcal{L}$ of nodes to be explored is initialized to contain only the root node.

While $\mathcal{L}$ is not empty, repeat the following four steps:

Node Selection Select and remove the best-first node from $\mathcal{L}$.

Updating If the optimal solution of relaxation problem (SDP-RLT) is ϵ_r-feasible, then update the incumbent solution and incumbent value. Otherwise, go to the branching step if the relaxation problem is feasible and its optimal objective value is less than the incumbent value minus ϵ_z.

Branching Obtain the branching variable x_i. Add two nodes to $\mathcal{L}$ corresponding to both children.

Cutting Add linear and polar cuts if possible by solving problem (LPolar) and (QPolar), respectively. Choose the branching value v_i for breaking the current interval and introduce the disjunctive cut to the relaxation problems corresponding to both children.

This is indeed a branch-and-cut algorithm in the sense that the cuts introduced at any node of the tree are valid everywhere in the subtree rooted at this node. The next theorem shows the finiteness and correctness of the proposed algorithm.

Theorem 5.4.1. *The proposed algorithm finds an ϵ_r-ϵ_z-optimal solution of problem* (CP) *in a finite number of iterations.*

Proof. First, we show that it takes finitely many steps to enumerate all ϵ_r-feasible solutions of problem (CP). Consider a node in the list with the bound on variable x_i being $[l_i,\ u_i]$. Let $v_i \in (l_i,\ u_i)$ be the point where the cutting step bisects the interval. We now show that the disjunctive cuts added into both children of the current node eliminate a non-negligible region of the relaxed domain. For the node where $l_i \le x_i \le v_i$, if $x_i \ge v_i - \frac{\epsilon_r}{u_i}$, then the maximal error of $X_{ii} - x_i^2$ is

$$(l_i + v_i)(v_i - \frac{\epsilon_r}{u_i}) - l_i v_i - (v_i - \frac{\epsilon_r}{u_i})^2 = \frac{v_i \epsilon_r}{u_i} - \frac{l_i \epsilon_r}{u_i} - \frac{\epsilon_r^2}{u_i^2} \le \frac{v_i \epsilon_r}{u_i} \le \epsilon_r.$$

Similarly, for the node where $v_i \le x_i \le u_i$, if $x_i \le v_i + \frac{\epsilon_r}{u_i}$, then the maximal error of $X_{ii} - x_i^2$ is

$$(u_i + v_i)(v_i + \frac{\epsilon_r}{u_i}) - u_i v_i - (v_i + \frac{\epsilon_r}{u_i})^2 = \epsilon_r - \frac{v_i \epsilon_r}{u_i} - \frac{\epsilon_r^2}{u_i^2} \le \epsilon_r.$$

Thus, the variable (x_i, X_{ii}) is within the tolerance if the interval length is less than $\frac{\epsilon_r}{u_i}$. If the relaxed solution falling into that domain is generated, then this interval will not be refined anymore. Therefore, there can only be finite steps for branching each variable. Theorems 5.3.1 and 5.3.2 imply that the linear and quadratic polar cuts are valid inequalities and the optimal solution (within the ϵ_z tolerance) is never eliminated from the list. It follows that there exists a node in the tree where an ϵ_r-ϵ_z-optimal solution will be identified. Since the enumeration stage for generating all ϵ_r-feasible solutions stops in finitely many steps, the proposed algorithm finds an ϵ_r-ϵ_z-optimal solution of problem (CP) in a finite number of iterations. □

5.5 Numerical Examples

In this section, the proposed algorithm is tested on some examples appeared in the literature. The algorithm is coded in MATLAB 2012b using software package CVX [54] to solve semidefinite programming programs. Computational experiments are conducted on a Windows PC using 2.40GHz dual core and 4.00GB memory.

Example 5.5.1. This example is a quadratic reformulation of a fourth degree polynomial problem in [17]. It is restated in [13], [57] and [60]. The reformulation is shown as follows.

$$
\begin{aligned}
\min \quad & x_3 + x_1x_5 + x_2x_5 + x_3x_5 \\
\text{s.t.} \quad & x_5 - x_1x_4 = 0, \\
& x_6 - x_2x_3 = 0, \\
& x_1^2 + x_2^2 + x_3^2 + x_4^2 = 40, \\
& x_5x_6 \geq 25, \\
& 1 \leq x_1 \leq 5, \quad 1 \leq x_3 \leq 5, \\
& 1 \leq x_2 \leq 5, \quad 1 \leq x_4 \leq 5.
\end{aligned}
$$

Using the precision $\epsilon_r = \epsilon_z = 10^{-6}$, the pre-processing phase takes 1.62 seconds to improve the bounds on x_5 and x_6 to $[1.1452, 21.3710]$, then the enumeration phase finds the solution

$$x^* = (1.0000,\ 4.7430,\ 3.8212,\ 1.3794,\ 1.3794,\ 18.1237)$$

with the objective value of 17.0140 in 4.33 seconds by exploring a total of 9 nodes. The total time to solve this problem is 14.60 seconds. Table 5.1 lists the comparison between the proposed algorithm and the one in [13]. The third column "No Polar Cuts" shows the results of our algorithm without using the polar cuts. From Table 5.1, we can see that the number of explored

Table 5.1: Comparison between *BCA-BQCQP* and Audet [13]

	Audet [13]	*BCA-BQCQP*	No Polar Cuts
Explored Nodes	69	9	213
Pre-Processing Phase	2.69	1.62	1.62
Enumeration Phase	4.49	4.33	149.61
Total CPU Time	7.18	5.95	151.23
Optimal Objective Value	17.0140	17.0140	17.0140

nodes by our algorithm is much smaller that of [13] if the polar cuts are applied. Without the polar cuts, the number of explored nodes increases dramatically.

Example 5.5.2. The following example is from [60].

$$\begin{aligned}
\min \quad & x_1 + x_2 + x_3 \\
\text{s.t.} \quad & 0.0025x_4 + 0.0025x_6 \leq 1, \\
& -0.0025x_4 + 0.0025x_5 + 0.0025x_7 \leq 1, \\
& -0.01x_5 + 0.01x_8 \leq 1, \\
& 100x_1 - x_1x_6 + 833.33x_4 \leq 83333.33, \\
& x_2x_4 - x_2x_7 - 1250x_4 + 1250x_5 \leq 0, \\
& x_3x_5 - x_3x_8 - 2500x_5 \leq -1250000, \\
& 100 \leq x_1 \leq 10000, \ 1000 \leq x_2, x_3 \leq 10000, \\
& 10 \leq x_4, x_5, x_6, x_7, x_8 \leq 1000.
\end{aligned}$$

Linderoth [72] solved this problem using a simplicial branch-and-bound algorithm. Our algorithm takes 12.47 seconds in the pre-processing phase to improve the bounds on x_4, x_5, x_6, x_7 and x_8 to [10, 395.6012]. Then, it takes 4.37 seconds in the enumerate phase to find the following solution with precision $\epsilon_r = \epsilon_z = 10^{-6}$:

$$x^* = (579.3067\ 1359.9707,\ 5109.9707,\ 182.0177,\ 295.6012,\ 217.9823,\ 286.4165,\ 395.6012)$$

The optimal value is 7049.2480, which is consistent with the one reported in GAMS GlobalLib [51]. Table 5.2 compares the number of nodes explored and the CPU time of our algorithm and the one in [72]. The third column "No Polar Cuts" shows the results of our algorithm without using the polar cuts. From Table 5.2, we can see that our algorithm outperforms the one in [72]

Table 5.2: Comparison between *BCA-BQCQP* and Linderoth [72]

	Linderoth[72]	*BCA-BQCQP*	No Polar Cuts
Explored Nodes	8810	3	177
Pre-Processing Phase	-	9.51	9.51
Enumeration Phase	28.92	4.91	302.32
Total CPU Time	28.92	14.42	311.83
Optimal Objective Value	7049.2480	7049.2480	7049.2480

by exploring less nodes and spending less CPU time in total. If the polar cuts are not used in the proposed algorithm, the number of explored nodes and the total CPU time increases a lot.

The proposed algorithm also solved several optimization problems from the literature. The QCQP formulations of these problems are detailed in [12].

Table 5.3 presents some important characteristics of these test problems. The first column

Table 5.3: Characteristics of the test problems

Id	Source	n	m
1	Haverly [58]	5	7
2	Colville [38]	4	9
3	Avriel and Williams [14]	4	7
4	Bracken and McCormick [29]	7	18
5	Dembo [43]	10	32
6	Dembo [43]	16	47

indicates the identification of each problem, the second column shows the reference in which these test problems appeared, and the last two columns quantify the size of the test problems.

The first test problem stems from the bilinear pooling problem encountered in the petrochemical industry. The next one describes a situation at *Proctor and Gamble Company*. The third one is a simple 3-stage heat exchange design problem. The fourth one models an alkylation process. The last two arise from membrane separation in three and five phases respectively. The diversity of these applications indicates the modeling flexibility of (BQCQP).

Table 5.4: Performance comparison between *BCA-BQCQP* and Audet [13]

Id	Audet [13]				*BCA-BQCQP*				$\epsilon_r = \epsilon_z$
	Nodes	PP	Tree	Tot	Nodes	PP	Tree	Tot	
1	9	0.64	0.01	0.65	3	0.97	2.72	3.69	10^{-6}
2	7	1.25	0.04	1.29	3	1.16	2.29	3.45	10^{-6}
3	191	1.1	3.8	4.9	5	1.1	4.4	5.5	10^{-6}
4	357	4.3	65.6	69.9	3	4.3	25.1	29.4	10^{-5}
5	259	148	61	209	7	146	20	166	10^{-5}
6	2847	222	7329	7551	7	221	31	252	10^{-5}

Table 5.4 displays the results of the proposed algorithm, comparing with the ones in [13]. The column “Nodes” shows the number of explored nodes in the enumeration phase. The columns “PP”, “Tree” and “Tot” indicate the pre-processing, enumeration and total CPU time, respectively, in seconds used by the corresponding algorithm. The last column displays the tolerance parameters supplied to the algorithm.

BCA-BQCQP solved all test problems for global optimality within a good precision by exploring less than 10 nodes. Especially, the proposed algorithm slightly improved the solution of problem 5 in shorter computational time, resulting an optimal objective value of 97.5372 instead of 97.5875. Based on the results of Table 5.4, we can see that the proposed algorithm becomes competitive when the problem size is large.

5.6 Conclusion

In this chapter, a new branch-and-cut algorithm for solving the bounded quadratically constrained quadratic programming problem has been developed. Two new classes of cuts, the generalized linear and quadratic polar cuts, have been derived to improve the bounds obtained at each node in the algorithm. In order to speed up the computation, a special branch-and-cut rule is adopted to enhance the approximation of the original feasible domain. Numerical examples have shown that the proposed branch-and-cut algorithm finds an approximate solution by exploring only a small number of nodes.

Chapter 6

Conclusions

In this dissertation, we have studied three important subclasses of quadratically constrained quadratic programming problems. We first summarize the results obtained in Section 6.1 and then suggest some directions for future research in Section 6.2.

6.1 Summary of Dissertation

The study of QCQP problems has lasted for several decades. QCQP problems form an important subclass of optimization problems in both theory and practice. On the theoretical side, the study of QCQP problems can help us understand the difference between convex and nonconvex problems in terms of computational difficulties. On the practical side, QCQP models have been widely adopted in real-life applications. A general QCQP problem is NP-hard, but some subclasses of QCQP problems can be solved efficiently by approximation algorithms. In this dissertation, three important subclasses of QCQP problems have been investigated: the standard quadratic programming (StQP) problem, the extended trust-region subproblem (ETRS) and the bounded quadratically constrained quadratic programming (BQCQP) problem. By exploring efficient approximation methods for these subclasses, we developed deeper understanding of the complicated structure of the general QCQP problem.

In Chapter 3, we have studied the problem (StQP). We first reformulated the problem as a linear conic programming problem on the cone of nonnegative quadratic functions over the standard simplex. A sequence of computable cones of nonnegative quadratic functions over a union of ellipsoids was used to approximate the cone of nonnegative quadratic functions over the standard simplex. In order to speed up the convergence of approximation and to relieve the computational burden, an adaptive scheme was adopted to refine the union of ellipsoids. Based on this scheme, we provided an iterative algorithm to detect the copositivity of a given matrix.

In Chapter 4, we have studied the problem (ETRS). This problem is also transformed into a linear conic programming problem on the cone of nonnegative quadratic functions over the feasible domain of problem (ETRS). A similar iterative algorithm were obtained based on a revised adaptive scheme. Moreover, the reformulation-linearization techniques (RLT) were applied to the adaptive scheme to further improve the quality of solutions obtained. If the feasible domain is bounded with a nonempty interior, the proposed algorithm has been proven to be able to find an ϵ-optimal solution within a finite number of iterations for any given small tolerance $\epsilon > 0$.

In Chapter 5, we have studied the problem (BQCQP). The conic reformulation and several convex relaxations were derived for the problem. In order to improve the relaxations, we developed generalized linear and quadratic polar cuts. A branch-and-cut algorithm based on these new cuts was then proposed with an adaptive branch-and-cut rule embedded. It was proven that the proposed algorithm yielded an ϵ_r-ϵ_z-optimal solution in a finite number of iterations.

6.2 Future Research

In this dissertation, we have studied three subclasses of QCQP problems based on the linear conic programming framework using the cone of nonnegative quadratic functions. The main difficulty in designing an efficient algorithm is to find the LMI representations of the cone of nonnegative quadratic functions over the feasible domain. Due to the NP-hardness of QCQP problems, it is not likely to find a computable LMI representation for every problem instance. We have derived an LMI representation for quadratic functions over a union of ellipsoids, which in turn was used for approximation of some NP-hard optimization problems like StQP and ETRS. Moreover, we derived generalized polar cuts, which were embedded in a branch-and-cut algorithm for solving the BQCQP problem. The findings in this dissertation lead us to some directions for future research as follows.

First, a computable representation of the cone of nonnegative quadratic functions over a domain can lead to a polynomial-time algorithm for solving the quadratic programming over that domain. To the best of our knowledge, the list of domains having computable representations only includes the domains determined by the following: (a) one quadratic inequality constraint [117]; (b) by one strictly convex/concave quadratic equality constraint [125]; (c) by one convex quadratic inequality and one linear inequality [117]; (d) by one elliptic constraint and two parallel linear constraints [36]; (e) by two convex quadratic inequalities with the same quadratic term [125]; or (f) by a second-order cone constraint with or without special linear constraints [66]. Up to now, when the domain is determined by two elliptic constraints, or by one elliptic constraints with two general linear inequality constraints, no results are known. Our future work is to explore the LMI representations of the cone of nonnegative quadratic functions over these potential domains. If we can provide computable representations, then linear conic programming can lead to new polynomial-time solvable subclasses of QCQP problems.

Second, the polar cuts in Chapter 5 have shown the advantages in refining the bounds of the BQCQP problem. Exploring other cuts that are computationally cheap to be derived would further improve the performance of our approximation algorithms. One possible way to generate such cuts is applying RLT to valid nonconvex quadratic inequalities.

Third, the linear conic programming framework studied in this dissertation has many advantages for solving QCQP problems. There is a possibility for us to extend this framework to study the polynomial optimization problems, which is even more complicated than QCQP problems.

REFERENCES

[1] Adams, W. P. and H.D. Sherali (1986). A tight linearization and an algorithm for zero-one quadratic programming problems. *Management Science*, 32(10), 1274-1290.

[2] Adams, W. P. and H.D. Sherali (1990). Linearization strategies for a class of zero-one mixed integer programming problems. *Operations Research*, 38(2), 217-226.

[3] Adams, W. P. and H.D. Sherali (1993). Mixed-integer bilinear programming problems. *Mathematical Programming*, 59(3), 279-306.

[4] Ai, W. and S. Zhang (2009). Strong duality for the CDT Subproblem: A necessary and sufficient condition. *SIAM Journal on Optimization*, 19(4), 1735-1756.

[5] Ai, W., Y. Huang and S. Zhang (2011). New results on Hermitian matrix rank-one decomposition. *Mathematical Programming*, 128(1-2), 253-283.

[6] Alders, C.D. and V.A. Sposito (1978). Necessary optimality conditions for programming problems over cone domains. *Optimization*, 9(4), 515-521.

[7] Alders, C.D. and V.A. Sposito (1979). A note on "real and complex Fritz John Theorems". *Journal of Mathematical Analysis and Applications*, 67(1), 92-93.

[8] Alizadeh, F., J.-P. Haeberly and M. Overton (1998). Primal-dual interior-point methods for semidefinite programming: convergence rates, stability and numerical results. *SIAM Journal on Optimization* 8(3), 746-768.

[9] Al-Khayyal, F.A. and T. Van Voorhis (1995). A relaxation method for nonconvex quadratically constrained quadratic programs. *Journal of Global Optimization*, 6(3), 215-230.

[10] Andersson, L.E., G. Chang and T. Elfving (1995). Criteria for copositive matrices using simplices and barycentric coordinates. *Linear Algebra and its Applications*, 220,9-30.

[11] Anstreicher, K.M. (2009). Semidefinite programming versus the reformulation-linearization technique for nonconvex quadratically constrained quadratic programming. *Journal of Global Optimization*, 43(2-3), 471-484.

[12] Audet, C. (1997). Optimisation globale structurée: propriétés, équivalences et résolution. Thèse de Doctorat, École Polytechnique de Montréal. Available at <http://www.gerad.ca/Charles.Audet/These.pdf>

[13] Audet, C., P. Hansen, B. Jaumard and G. Savard (2000). A branch and cut algorithm for nonconvex quadratically constrained quadratic programming. *Mathematical Programming*, 87(1), 131-152.

[14] Avriel, M. and A.C. Williams. (1971). An extension of geometric programming with applications in engineering optimization. *Journal of Engineering Mathematics*, 5(2), 187-194.

[15] Balas, E. (1971). Intersection cuts – a new type of cutting planes for integer programming. *Operations Research*, 19(1), 19-39.

[16] Balas, E. (1979). Disjunctive programming. *Annals of Discrete Mathematics*, 5(1), 3-51.

[17] Bartholomew-Biggs, M.C. (1976) *A numerical comparison between two approaches to the nonlinear programming problem, Technical Report #77*, Numericlal Optimization Center, Hartfield Polytechnic, England.

[18] Beck, A. and M. Teboulle (2000). Global optimality conditions for quadratic optimization problems with binary constraints. *SIAM Journal on Optimization*, 11(1), 179-188.

[19] Beck, A. and Y.C. Eldar (2007). Strong duality in nonconvex quadratic optimization with two quadratic constraints. *SIAM Journal on Optimization*, 17(3), 844-860.

[20] Ben-Tal, A. and A. Nemirovskii (2001). *Lectures on Modern Convex Optimization: Analysis, Algorithms and Engineering Applications*. Philadelphia, PA: Society for Industrial and Applied Mathematics.

[21] Bomze, I.M. (2000). Linear-time copositivity detection for tridiagonal matrices and extension to block-tridiagonality. *SIAM Journal on Matrix Analysis and Applications*, 21(3), 840-848.

[22] Bomze, I.M., M. Dür, E. de Klerk, C. Roos, A.J. Quist and T. Terlaky (2000). On copositive programming and standard quadratic optimization problems. *Journal of Global Optimization*, 18(4), 301-320.

[23] Bomze, I.M. and E. de Klerk (2002). Solving standard quadratic optimization problems via linear, semidefinite and copositive programming. *Journal of Global Optimization*, 24(2), 163-185.

[24] Bomze, I.M., M. Locatelli and F. Tardella (2008). New and old bounds for standard quadratic optimization: dominance, equivalence and incomparability. *Mathematical Programming*, 115(1), 31-64.

[25] Bomze, I.M. and G. Eichfelder (2012). Copositivitity detection by difference-of-convex decomposition and ω-subdivision. *Mathematical Programmig*, 138(1-2), 365-400.

[26] Bose, S., D. Gayme, K. Chandy and S. Low (2012). Quadratically constrained quadratic programs on acyclic graphs with application to power flow. Available at <http://arxiv.org/pdf/1203.5599v2.pdf>

[27] Boyd, S., L.E. Ghaoui, E. Feron, and V. Balakrishnan (1994). *Linear Matrix Inequailities in System and Control Theory*. Philadelphia, PA: Society for Industrial and Applied Mathematics.

[28] Boyd, S. and L. Vandenberghe (2004). *Convex Optimization*. Cambridge, UK: Cambridge University Press.

[29] Bracken, J. and G.P. McCormick (1968). *Selected Applications of Nonlinear Programming.* New York, NY: John Wiley & Sons Press.

[30] Bremner D. , K. Fukuda and A. Marzetta (1998). Primal-dual methods for vertex and facet enumeration. *Discrete and Computational Geometry*, 20(3), 333-357.

[31] Bundfuss, S. and M. Dür (2008). Algorithmic copositivity detection by simplicial partition. *Linear Algebra and its Applications*, 428(7), 1511-1523.

[32] Bundfuss, S. and M. Dür (2009). An adaptive linear approximation algorithm for copositive programs. *SIAM Journal on Optimization*, 20(1), 30-53.

[33] Buer, S. and K.M. Anstreicher (2008). A finite branch-and-bound algorithm for nonconvex quadratic programming via semidefinite relaxations. *Mathematical Programming*, 113(2), 259-282.

[34] Burer, S. (2009). On the copositive representation of binary and continuous nonconvex quadratic programs. *Mathematical Programming*, 120(2), 479-495.

[35] Burer, S. and A. Saxena (2009). *Old wine in a new bottle: the MILP road to MIQCP*, Technical Report, Department of Management Sciences, University of Iowa. Available at <http://www.optimization-online.org/DB_FILE/2009/07/2338.pdf>

[36] Burer, S. and K.M. Anstreicher (2013). Second-order-cone constraint for extended trust-region subproblems. *SIAM Journal on Optimization*, 23(1), 432-451.

[37] Celis, M.R., J.E. Dennis and R.A. Tapia (1985). A trust region strategy for nonliear equality constrained optimization. In: P.T. Boggs et al. (Eds.), *Numerical Optimzation 1984, Proceedings of the SIAM Conference on Numerical Optimization*, 71-82, Boulder, CO.

[38] Colville, A.R. (1970). A comparative study of nonlinear programming codes. In: H.W. Kuhn (Eds), *Princeton Symposium on Mathematical Programming*, 293-312, Princetion University Press, NY.

[39] Conn, A.R., N.I.M. Gould and P.L. Toint (2000). *Trust-region Methods.* Philadelphia, PA: Society for Industrial and Applied Mathematics.

[40] Craven, B.D. and B. Mond (1973). Real and complex Fritz John Theorems. *Journal of Mathematical Analysis and Applications*, 44(3), 773-778.

[41] Czyzyk, J., M. Mesnier and J. Moré (1998). The NEOS Server. *Journal IEEE Computational Science and Engineering*, 5(3), 68-75.

[42] Danniger, G. (1990). A recursive algorithm for determing (strict) copositivity of a symmetric matrix. In: U. Rieder et al. (Eds.), *Methods of Operations Research 62*, 45-52, Hain, Meisenheim.

[43] Dembo, R.S. (1976). A set of geometric programming test problems and their solutions. *Mathematical Programming*, 10(1), 192-213.

[44] Deng, Z., S.-C. Fang, Q. Jin and C. Lu (2012). Conic approximation to nonconvex quadratic programming with convex quadratic constraints. Submitted to *Journal of Global Optimization.*

[45] Deng, Z., S.-C. Fang, Q. Jin and W. Xing (2013). Detecting copositivity of a symmetric matrix by an adaptive ellipsoid-based approximation scheme. *European Journal of Operational Research*, 229(1), 21-28.

[46] Deng, Z., S.-C. Fang, Y. Tian and W. Xing (2013). A branch-and-cut algorithm for nonconvex quadratically constrained quadratic programming based on linear and quadratic polar cuts. Working paper.

[47] Dukanovic, I. and F. Rendl (2009). Copositive programming motivated bounds on the stability and the chromatic numbers. *Mathematical Programming*, 121(2), 249-268.

[48] Fang, S.-C. and S. Puthenpura (1993). *Linear Optimization and Extensions: Theory and Algorithms.* Englewood Cliffs, NJ: Prentice-Hall.

[49] Fujie, T. and M. Kojima (1997). Semidefinite programming relaxation for nonconvex quadratic programs. *Journal of Global Optimization*, 10, 367-380.

[50] Gahinet P. and A. Nemirovsky (1993). *LMI: A Package for Manipulating and Solving LMIs*. National Institute for Research in Computer Science and Control.

[51] GlobalLib (2004). Available at <http://www.gamsworld.org/global/globallib.htm>

[52] Golub, G.H. and C. F. Van Loan (1996). *Matrix Computations.* Baltimore, MD: John Hopkins University Press.

[53] Gould, N.I.M., S. Lucidi, M. Roma and P. L. Toint (1999). Solving the trust-region subproblem using the Lanczos method. *SIAM Journal on Optimization*, 9(2), 504-525.

[54] Grant, M. and S. Boyd (2011). CVX: Matlab software for disciplined convex programming, version 2.00 beta. Available at <http://cvxr.com/cvx/>

[55] Gröschel, M., L. Lovasz and A. Schrijver (1998). *Geometric Algorithms and Combinatorial Optimization.* Berlin, Germany: Springer-Verlag Press.

[56] Gvozdenović, N. and M. Laurent (2008). The operator Ψ for the chromatic number of a graph. *SIAM Journal on Optimization*, 19(2), 572-591.

[57] Hansen, P. and B. Jaumard (1992). Reduction of indefinite quadratic programs to bilinear programs. *Journal of Global Optimization*, 2(1), 41-60.

[58] Haverly, C.A. (1978). Studies of the behaviour of recursion for the pooling problem. *ACM SIGMAP Bulletin*, 25, 19-28.

[59] Hiriart-Urruty, J.-B. and A. Seeger (2010). A variational approach to copositive matrices. *SIAM Review*, 52(4), 593-629.

[60] Hock, W. and K. Schittkowski (1981). *Text Examples for Nonlinear Programming Codes, Lecture Notes in Economics and Mathematical Systems 187*, Berlin, DE: Springer.

[61] Horn, R.A. and C.R. Johnson (1990). *Matrix Analysis.* Cambridge, UK: Cambridge University Press.

[62] Horst, R. and N.V. Thoai (1996). A new algorithm for solving the general quadratic programming problem. *Computational Optimization and Applications*, 5, 39-48.

[63] Huang, Y. and S. Zhang (2007). Complex matrix decomposition and quadratic programming. *Mathematics of Operations Research*, 32(3), 758-768.

[64] Ikramov, K.D. (2002). A linear-time algorithm for verifying the copositivity of an acyclic matrix. *Computational Mathematics and Mathmetical Physics*, 42(12), 1701-1703.

[65] Jeyakumar, V., A.M. Rubinov and Z. Wu (2007). Nonconvex quadratic minimization problems with quadratic constraints: global optimality conditions. *Mathematical Programming*, 110(3), 521-541.

[66] Jin, Q., Y. Tian, Z. Deng, S.-C. Fang and W. Xing (2013). Exact computable representation of some second-order cone constrained quadratic programming problems. *Journal of Operations Research Society of China*, 1(1), 107-134.

[67] Kaplan, W. (2001). A copositivity probe. *Linear Algebra and its Applications*, 337(1-3), 237-251.

[68] Kim, S. and M. Kojima (2001). Second order cone programming relaxation of nonconvex quadratic optimization problems. *Optimization Methods and Software*, 15(3-4), 201-224.

[69] de Klerk, E. and D.V. Pasechnik (2002). Approximation of the stability number of a graph via copositive programming. *SIAM Journal on Optimization*, 12(4), 875-892.

[70] de Klerk, E. and D.V. Pasechnik (2007). A linear programming reformulation of the standard quadratic optimization problem. *Journal of Global Optimization*, 37(1), 75-84.

[71] Kuhn, H.W. and A.W. Tucker (1951). Nonlinear programming. In: J. Neyman (Eds.), *Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability*, 481-492. Berkeley, CA.

[72] Linderoth, J. (2005). A simplicial branch-and-bound algorithm for solving quadratically constrained quadratic programs. *Mathematical Programming*, 103(2), 251-282.

[73] Lu, C., S.-C. Fang, Q. Jin, Z. Wang and W. Xing (2010). KKT solution and conic relaxation for solving quadratically constrained quadratic programming problems. *SIAM Journal on Optimization*, 21(4), 1475-1490.

[74] Lu, C., Q. Jin, S.-C. Fang and W. Xing (2011). LMI based adaptive approximation strategy to cones of nonnegative quadratic functions. To appear in *Optimization*.

[75] Luo, Z.-Q., J.F. Sturm and S. Zhang (2004). Multivariate nonnegative quadratic mappings. *SIAM Journal on Optimization*, 14(4), 1140-1162.

[76] Lur'e A.I. (1957). *Some Nonlinear Problems in the Theory of Automatic Control.* London, UK: H.M. Stationery Office.

[77] de Maio, Antonion, Y. Huang, D.P. Palomar, S. Zhang and A. Farinan (2011). Fractional QCQP with applications in ML steering direction estimation for radar detection. *IEEE Transactions on Signal Processing*, 59(1), 172-185.

[78] Matsui, T. (1996). NP-hardness of linear multiplicative programming and related problems. *Journal of Global Optimization*, 9(2), 113-119.

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

[80] Motzkin, T.S. (1952). Copositive quadratic forms. *National Bureau of Standards Report 1818*, 11-22.

[81] Motzkin, T.S. and E.G. Straus (1965). Maxima for graphs and a new proof of a theorem of Turán. *Canadian Journal of Mathmatics*, 17, 533-540.

[82] Murty, K.G. and S.N. Kabadi (1987). Some NP-Complete problems in quadratic and nonlinear programming. *Mathematical Programming*, 39(2), 117-129.

[83] Nazareth, J.L. (1991). The homotopy principle and algorithms for linear programming. *SIAM Journal on Optimization*, 1(3), 316-332.

[84] Nesterov, Y. E. and A.S. Nemirovskii (1994). *Interior-point Polynomial Algorithms in Convex Programming.* Philadelphia, PA: Society for Industrial and Applied Mathematics.

[85] Pardalos, P.M. and S.A. Vavasis (1991). Quadratic programming with one negative eigenvalue is NP-Hard. *Journal of Global Optimization*, 1(1), 15-22.

[86] Parida, J. and K.L. Roy (1982). An existence theorem for quadratic programs over cone domains. *Journal of Applied Mathematics and Mechanics*, 62(3), 207-208.

[87] Parrilo, P.A. (2000). *Structured Semidefinite Programs and Semialgebraic Geometry Methods in Robustness and Optimization.* PhD thesis, California Institute of Technology, CA.

[88] Peña, J., J. Vera and L. Zuluaga (2007). Computing the stablility number of a graph via linear and semidefinite programming. *SIAM Journal on Optimization*, 18(1), 87-105.

[89] Poljak, S., F. Rendl and H. Wolkowicz (1995). A recipe for semidefinite relaxation for (0,1)-quadratic programming. *Journal of Global Optimization*, 7(1), 51-73.

[90] Povh, J. and F. Rendl (2007). A copositive programming approach to graph partitioning. *SIAM Journal on Optimization*, 18(1), 223-241.

[91] Povh, J. and F. Rendl (2009). Copositive and semidefinite relaxations of the quadratic assignment problem. *Discrete Optimization*, 6(3), 231-241.

[92] Preisig, J.C. (1996). Copositivity and minimization of quadratic functions with nonnegativity and quadratic equality constraints. *SIAM Journal on Control and Optimization*, 34(4), 1135-1150.

[93] Quist, A.J., E. de Klerk, C. Roos and T. Terlaky (1998). Copositive relaxation for genreal quadratic programming. *Optimization Methods and Software*, 9(1-3), 185-208.

[94] Raber, U. (1998). A simplicial branch-and-bound method for solving nonconvex all-quadratic programs. *Journal of Global Optimization*, 13(4), 417-432.

[95] Rendl, F. and H. Wolkowicz (1997). A semidefinite framework for trust region subproblems with applications to large scale minimization. *Mathematical Programming*, 77(2), 273-299.

[96] Rockafellar, R.T. (1972). *Convex Analysis* (second edition). Princeton, NJ: Princeton University Press.

[97] Sahinidis, N.V. (1996). BARON: a general purpose global optimization software package. *Journal of Global Optimization*, 8(2), 201-205.

[98] Saxena, A., P. Bonami and J. Lee (2010). Convex relaxations of non-convex mixed integer quadratically constrained programs: extended formulations. *Mathematical Programming*, 124(1-2), 383-411.

[99] Saxena, A., P. Bonami and J. Lee (2011). Convex relaxations of non-convex mixed integer quadratically constrained programs: projected formulation. *Mathematical Programming*, 130(2), 359-413.

[100] Sherali, H.D. and W.P. Adams (1990). A hierarchy of relaxations between the continuous and convex hull representations for zero-one programming problems. *SIAM Journal on Discrete Mathematics*, 3(3), 411-430.

[101] Sherali, H.D. and W.P. Adams (1994). A hierarchy of relaxations and convex hull characterizations for mixed-integer zero-one programming problems. *Discrete Applied Mathematics*, 52(1), 83-106.

[102] Sherali, H.D. and A.R. Alameddine (1992). A new reformulation-linearization technique for bilinear programming problems. *Journal of Global Optimization*, 2(4), 379-410.

[103] Sherali, H.D. and C.H. Tuncbilek (1992). A global optimization algorithm for polynomial programming problems using a reformulation-linearization technique. *Journal of Global Optimization*, 2(1), 101-112.

[104] Sherali, H.D. and C.H. Tuncbilek (1995). A reformulation-convexification approach for solving nonconvex quadratic programming problems. *Journal of Global Optimization*, 7(1), 1-31.

[105] Sherali, H.D. and C.H. Tuncbilek (1997). New reformulation linearization/convexification relaxations for univariate and multivaraite polynomial programming problems. *Operations Research Letters*, 21(1), 1-9.

[106] Sherali, H.D. and W.P. Adams (1999). *A Reformulated-Linerization Technique for Solving Discrete and Continuous Nonconvex Problems.* Dordrecht, NL: Kluwer Academic Publishers.

[107] Sherali, H.D. and H. Wang (2001). Global optimization of nonconvex factorable programming problems. *Mathematical Programming*, 89(3), 459478.

[108] Sherali, H.D. (2007). RLT: A unified approach for discrete and continuous nonconvex optimization. *Annals of Operations Research*, 149(1), 185-193.

[109] Sherali, H.D. and L. Liberti (2009). Reformulation-linearization methods for global optimization. In F. Christodoulos and P. Panos (Eds.), *Encyclopedia of Optimization* (second edition), 3263-3268, New York, NY: Springer.

[110] Shor, N.Z. (1987). Quadratic optimization problems. *Soviet Journal of Computer and Systems Sciences*, 25, 1-11.

[111] Signoretto, M., K. Pelckmans and J. Suykens (2008). *Quadratically constrained quadratic programming for subspace selection in kernel regression estimation*, in: Artificial Neural Networks–ICANN 2008, 175-184.

[112] Sposito, V.A. and H.T. David (1971). Saddle-point optimality criteria of nonlinear programming problems over cones without differentiability. *SIAM Journal on Applied Mathematics*, 20(4), 698-702.

[113] Sposito, V.A. and H.T. David (1972). A note on Farkas Lemmas over cone domains. *SIAM Journal on Applied Mathematics*, 22(3), 356-358.

[114] Sposito, V.A. (1974). Modified regularity conditions for nonlinear programming problems over mixed cone domain. *Mathematical Programming*, 6(1), 167-179.

[115] Sposito, V.A. (1976). Quadratic duality over convex cone domains. *Mathematical Programming*, 10(1), 277-283.

[116] Sturm, J.F. (1999). Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. *Optimization Methods and Software*, 11(1-4), 625-653. Version 1.3 available from http://sedumi.ie.lehigh.edu/.

[117] Sturm, J.F. and S. Zhang (2003). On cones of nonnegative quadratic functions. *Mathematics of Operations Research*, 28(2), 246-267.

[118] Tian, Y., S.-C. Fang, Z. Deng and W. Xing (2013). Copositive programming approximation by adaptive computable second order cones. *Journal of Industrial and Management Optimization*, 9(3), 703-721.

[119] Vanantwerp J.G. and R.D. Braatz (2000). A tutorial on linear and bilinear matrix inequalities. *Journal of Process Control*, 10(4), 363-385.

[120] Voorhis, T.V. (2002). A global optimization algorithm using Lagrangian underestimates and the interval Newton method. *Journal of Global Optimization*, 24(3), 349-370.

[121] Yakubovich V.A. (1962) The solution of certain matrix inequalities in automatic control theory. *Soviet Mamthematics Doklady*, 3, 620-623.

[122] Yakubovich V.A. (1967) The method of matrix inequalities in the stability theory of nonlinear control systems, I, II, III. *Automation and Remote Control*, 25-26, 905-917, 577-592, 753-763.

[123] Yang, S. and X. Li (2009). Algorithms for determining the copositivity of a given symmetric matrix. *Linear Algebra and Its Applications*, 430(2-3), 609-618.

[124] Ye, Y. (1997). *Interior Point Algorithms: Theory and Analysis*. New York, NY: John Wiley and Sons.

[125] Ye, Y. and S. Zhang (2003). New results on quadratic minimization. *SIAM Journal on Optimization*, 14(1), 245-267.

[126] Yıldırım, E.A. (2012). On the accuracy of uniform polyhedral approxiamtions of the copositive cone. *Optimization Methods and Software*, 27(1), 155-173.

[127] Zhang, S. (2000). Quadratic maximization and semidefinite relaxation. *Mathematical Programming*, 87(3), 453-465.

[128] Zheng, X.J., X.L. Sun and D. Li (2011) Nonconvex quadratically constrained quadratic programming: best D.C. decomposition and their SDP represetations. *Journal of Global Optimization*, 50(4), 695-712.

[129] Zheng, X.J., X.L. Sun and D. Li (2011) Convex relaxations for nonconvex quadratically constrained quadratic programming: matrix cone decomposition and polyhedral approximation. *Mathematical Programming*, 129(2), 301-329.

[130] Zhu, S., D. Li and X. Sun (2010). Portfolio selection with marginal risk control. *Journal of Computational Finance*, 14(1), 3-28.

[131] Žilinskas, J. and M. Dür (2011). Depth-first simplicial partition for copositivity detection, with an application to MaxClique. *Optimization Methods and Software*, 26(3), 499-510.