Notation

$\mathbb{R}$ - Set of real numbers
$\mathbb{R}_+$ - Set of non-negative real numbers
$\mathbb{R}^n$ - Set of n-dimensional real vectors
$\mathbb{R}^n_+$ - Set of n-dimensional real vectors with non-negative entries

S^n - Unit sphere in $\mathbb{R}^n$

$\mathbb{Q}^n$ - Second order cone

$\mathbb{S}^n$ - Set of symmetric $n \times n$ matrices
$\mathbb{S}^n_+$ - Set of symmetric positive semidefinite $n \times n$ matrices (positive semidefinite cone)
$\mathbb{S}^n_{++}$ - Set of symmetric positive definite $n \times n$ matrices

$\mathbb{C}^n$ - Set of symmetric copositive $n \times n$ matrices (copositive cone)
$\mathbb{P}^n$ - Set of symmetric completely positive $n \times n$ matrices (completely positive cone)
$\mathbb{N}^n$ - Set of symmetric nonnegative $n \times n$ matrices (nonnegative cone)
$\mathbb{S}^n_+ \cap \mathbb{N}^n$ - Doubly nonnegative cone (DNN) Γ - Set of positive rank 1 matrices

$M \succ 0$ - M is symmetric and positive definite
$M \succeq 0$ - M is symmetric and positive semidefinite
$M_1 \succeq M_2$ - $M_1 - M2 \succeq 0$
$C \bullet X$ - Matrix inner product $= Tr(C^T X)$

$conv(K)$ - Convex hull of K
$svec(\cdot)$ - An operator transforming a symmetric $n \times n$ matrix into a vector vector of its entries in $\mathbb{R}^{n(n+1)/2}$.

PSD - Positive Semidefinite

QCQP - Quadratically Constrained Quadratic Program
LP - Linear Program
SOCP - Second Order Cone Program
SDP - Semidefinite program
CP - Copositive program
CPP - Completely positive program

Chapter 1

Introduction

In this thesis we will study quadratically constrained quadratic programs (QCQP)

Definition 1.1. The Quadratically Constrained Quadratic Program (QCQP) in the standard form is

$$\begin{array}{ll} \text{minimize} & x^T P_0 x + q_0^T x + r_0 \\ \text{subject to} & x^T P_k x + q_k^T x + r_k \leq 0, \ (k = 1, \ldots, m), \end{array} \tag{1.2}$$

where $x \in \mathbb{R}^n$ is a variable, and symmetric $n \times n$ matrices $P_0, P_1, \ldots, P_m \in S^n$, vectors $q_0, \ldots, q_m \in \mathbb{R}^n$ and scalars $r_1, \ldots, r_m \in \mathbb{R}$ are given.

The matrices $P_0, P_1, \ldots P_m$ are not necessarily positive semidefinite. Therefore the objective function as well as the constraints may be nonconvex. Problem (1.2) has been proved to be NP-hard in general [39], while several special subclasses of QCQP have been identified to be polynomially solvable (see [2]).

For example, the 0-1 constraint $x_i \in \{0, 1\}$ can be reformulated as $x^T e_i e_i^T x - e_i^T x = 0$. Thus QCQP includes 0-1 programming, which describes various NP-hard problems, such as knapsack, stable set, max cut etc. On the other hand, the above examples suggest that QCQP has many applications and is worth solving.

One of the possible approaches is relaxing QCQP to obtain convex problems which can be solved in polynomial time, namely linear programming (LP), second order cone programming (SOCP), or semidefinite programming (SDP).

Chapter 2

Conic optimization classes

In this section we will introduce basic optimization classes mentioned above, in particular the linear programming (LP), second order cone programming (SOCP), or semidefinite programming (SDP). We will state the problems in standard forms and their duals. The dual problems will be only mentioned here and will be derived in the next section. We will also show that LP is subclass of convex QCQP, convex QCQP is subclass of SOCP, and SOCP is subclass of SDP, i.e.

$$LP \subset \text{convex } QCQP \subset SOCP \subset SDP \subset QCQP. \tag{2.1}$$

2.1 Linear programming

When both, the objective and the constraint functions are linear (affine), the problem is called a linear program and it belongs to the Linear Programming class, or shortly LP. In this section we will introduce the standard form of LP and its dual. For reference and more information about this topic see i.e. [5] .

Definition 2.2. The primal–dual pair of linear programs in standard form is

$$\begin{array}{llcll} & \textit{Primal} & & & \textit{Dual} \\ \text{minimize} & c^T x, & & \text{maximize} & b^T y, \\ \text{subject to} & Ax = b, & & \text{subject to} & A^T y + s = c, \\ & x \in \mathbb{R}^n_+, & & & s \in \mathbb{R}^n_+, \end{array} \tag{2.3}$$

where $x \in R^n$, $y \in \mathbb{R}^m$, $s \in \mathbb{R}^n$ are the variables; the real $m \times n$ matrix A and vectors $b \in \mathbb{R}^m, c \in \mathbb{R}^n$ are given problem data.

Remark 2.4. Linear programs can be formulated in various forms (including $\geq, \leq$ inequalities, free variables, possibly some linear fractions in objective) but all of them can be transformed to the standard form.

2.2 Second order cone programing

The second order cone programming (SOCP) is a convex optimization class which can be solved with great efficiency using interior point methods. In this section we will introduce the standard form of SOCP and its dual. For reference and more information about this topic see [5] .

Let us first define second order cone.

Definition 2.5 (Second order cone)**.** We say $\mathbb{Q}^n$ is second order cone of dimension n if

$$\mathbb{Q}^n = \{x \in \mathbb{R}^n \mid x = (x_0, \bar{x}) \in \mathbb{R} \times \mathbb{R}^{n-1}, \|\bar{x}\|_2 \leq x_0\}. \tag{2.6}$$

Definition 2.7 (SOCP)**.** The primal–dual pair of the Second Order Cone Program (SOCP) in the standard form is

$$\begin{array}{llcll} & \textit{Primal} & & & \textit{Dual} \\ \text{minimize} & c^T x, & & \text{maximize} & b^T y, \\ \text{subject to} & Ax = b, & & \text{subject to} & A^T y + s = c, \\ & x \in \mathbb{Q}^n, & & & s \in \mathbb{Q}^n, \end{array} \tag{2.8}$$

where $x \in \mathbb{R}^n$, $y \in \mathbb{R}^m$ and $s \in \mathbb{R}^n$ are the variables; and $m \times n$ real matrix A, vectors $c \in \mathbb{R}^n$, $b \in \mathbb{R}^m$, $c \in \mathbb{R}^n$ are given problem data.

Remark 2.9. Second order cone programs can be formulated in various forms (including quadratic objective or several second order cone constraints of the affine functions), but all of them can be transformed to the standard form.

Remark 2.10. In general, any program of the form

$$\begin{array}{llcll} & \textit{Primal} & & & \textit{Dual} \\ \min & c^{1T} x^1 + \cdots + c^{kT} x^k, & & \max & b^T y, \\ \text{s.t.} & A^1 x^1 + \cdots + A^k x^k = b, & & \text{s.t.} & A^{iT} y + s^i = c^i, \\ & x^i \in \mathbb{Q}^{n_i},\ (i = 1, \ldots, k), & & & s^i \in \mathbb{Q}^{n_i},\ (i = 1, \ldots, k), \end{array} \tag{2.11}$$

is considered to be SOCP. The second order cone constraints can be also formulated as, $x = (x^1, \ldots, x^k) \in \mathbb{Q}$, where $\mathbb{Q}$ is Cartesian product of second order cones,

$$\mathbb{Q} = \mathbb{Q}^{n_1} \times \mathbb{Q}^{n_2} \times \cdots \times \mathbb{Q}^{n_k}, \tag{2.12}$$

such $\mathbb{Q}$ has all important properties of second order cone, and algorithmic aspects of solving standard SOCP can also be applied for this more general case [7,8]. In order to keep things simple, we will sometimes consider only the standard form stated in the Definition 2.7, but all the details can be also done for this more general form.

2.2.1 Relation to previous classes

Second Order Cone Programming includes convex LP as special case. We will show that SOCP in fact includes convex QP as a subclass. We will demonstrate procedure proposed in [4] used to reformulate convex QP as SOCP.

Let us have convex QCQP. In other words, suppose that $n \times n$ matrices A_k, $k = 0, \dots, m$ are positive semidefinite.

$$\begin{array}{ll} \text{minimize} & x^T P_0 x + q_0^T x + r_0, \\ \text{subject to} & x^T P_k x + q_k^T x + r_k \leq 0, \ (k = 1, \dots, m), \end{array} \tag{2.13}$$

First of all, rewrite problem equivalently as

$$\begin{array}{ll} \text{minimize} & t, \\ \text{subject to} & x^T P_0 x + q_0^T x + r_0 \leq t, \\ & x^T P_k x + q_k^T x + r_k \leq 0, \ (k = 1, \dots, m). \end{array} \tag{2.14}$$

To avoid tedious notation, without loss of generality, suppose that considered program already has linear objective function (i.e. $P_0 = 0$). Also suppose that we have separated all the linear constraints (ones where $P_k = 0$) and arrange them into more compact form $Ax = b$. Even if we did not, the following procedure will still be correct, but will result in more complicated formulation of linear constraints.

Each convex quadratic constraint

$$x^T P x + q^T x + r \leq 0 \tag{2.15}$$

can be transformed into the second order cone constraint. Suppose that $P \neq 0$ and rank $P = h$. Then there exists $n \times h$ matrix L such that $P = LL^T$. Such L can be computed by Choelsky factorization of P. Now rewrite (2.15) as

$$(L^T x)^T (L^T x) \leq -q^T x - r. \tag{2.16}$$

It can be easily verified that $w \in \mathbb{R}^t$, $\xi \in \mathbb{R}$ and $\eta \in \mathbb{R}$ satisfy

$$w^T w \leq \xi \eta, \quad \xi \geq 0 \ \text{ and } \ \eta \geq 0$$

if and only if they satisfy

$$\left\| \begin{pmatrix} \xi - \eta \\ 2w \end{pmatrix} \right\|_2 \leq \xi + \eta.$$

If we take $w = L^T x$, $\xi = 1$ and $\eta = -q^T x - r$, then inequality (2.16) is equivalent to the second order cone constraint

$$\|v\|_2 \leq v_0, \quad \text{where} \quad \begin{pmatrix} v_0 \\ v \end{pmatrix} = \begin{pmatrix} 1 - q^T x - r \\ 1 + q^T x + r \\ 2L^T x \end{pmatrix} \in \mathbb{R}^{h+2}. \tag{2.17}$$

Now the intersection of all such second order cone constraints can be easily expressed as Cartesian product of second order cones, thus we have obtained problem of SOCP in form 2.11.

2.3 Semidefinite programming

The semidefinite programming (SDP) is a convex optimization class which can be solved efficiently using interior point methods. In this section we will introduce the standard form of SDP and its dual. For reference and more information about this topic see [5] .

Firstly, let us introduce notation we will use to simplify the standard form.

Definition 2.18. Let A, X be real $n \times m$ matrices, we will denote their inner product

$$A \bullet X = Tr(A^T X).$$

Where $Tr(M)$ denotes trace of matrix M i.e. sum of the diagonal elements of M.

Definition 2.19 (SDP)**.** The primal–dual pair of the Semidefinite Program (SDP) in the standard form is

$$\begin{array}{ll} & \textit{Primal} \\ \text{minimize} & A_0 \bullet X, \\ \text{subject to} & A_k \bullet X = b_k, \\ & (k = 1, \ldots, m), \\ & X \in \mathbb{S}^n_+, \end{array} \qquad \begin{array}{ll} & \textit{Dual} \\ \text{maximize} & b^T y, \\ \text{subject to} & \sum_{k=1}^m y_k A_k + S = A_0, \\ & S \in \mathbb{S}^n_+, \end{array} \tag{2.20}$$

where $X \in \mathbb{S}^n_+$, $y = (y_1, \ldots, y_m)^T \in \mathbb{R}^m$ and $S \in \mathbb{S}^n$ are the variables; and symmetric matrices $A_0, A_1, \ldots, A_m \in \mathbb{S}^n$ and scalars $b_1, \ldots, b_m \in \mathbb{R}$ are given.

Surprisingly, the variable in SDP is a symmetric matrix (not a vector). In order to be consistent with other classes we will sometimes use a *svec* operator.

Definition 2.21. We define operator $svec : \mathbb{S}^n \to \mathbb{R}^{n(n+1)/2}$, such that for any $n \times n$ symmetric matrix M

$$svec(M) = (\delta_{11} M_{11}, \delta_{12} M_{12}, \delta_{22} M_{22}, \ldots, \delta_{1n} M_{1n}, \ldots, \delta_{nn} M_{nn})^T, \tag{2.22}$$

where

$$\delta_{ij} = \begin{cases} 1, & \text{if } \ i = j, \\ \sqrt{2}/2, & \text{otherwise.} \end{cases} \tag{2.23}$$

Notice that δ_{ij} are defined cleverly, so that inner product of symmetric matrices is equivalent to the standard inner product of their images

$$A \bullet X = svec(A)^T svec(X), \tag{2.24}$$

for any pair of symmetric matrices $A, X \in \mathbb{S}^n$. Now we can easily formulate the problems of SDP in terms of standard inner product over the space of real vectors.

$$\begin{array}{ll} \text{minimize} & svec(A_0)^T svec(X), \\ \text{subject to} & svec(A_k)^T svec(X) = b_k, \ (k = 1, \ldots, m), \\ & svec(X) \in \mathcal{K}(\mathbb{S}^n_+), \end{array} \tag{2.25}$$

where $svec(X)$ is variable, $\mathcal{K}(\mathbb{S}^n_+) = \{svec(U) \in \mathbb{R}^{n(n+1)/2} \mid U \in \mathbb{S}^n_+\}$ and problem data are from the standard SDP (2.20).

2.3.1 Relation to previous classes

The SDP primal-dual pair looks suspiciously similar to the both LP and SOCP primal-dual pairs. The only difference between LP and SOCP is the nonnegative orthant is replaced by second order cone. The SDP, in the *svec*-operator form (2.25), further generalizes the cone constraint with the semidefinite cone.

In fact, SOCP is subclass of SDP. We will show how the standard SOCP can be rewritten as SDP. First of all, instead of minimizing $c^T x$ we will minimize t with additional constraint $t \geq c^T x$.

The only nontrivial part is to rewrite conic constraint

$$x \in \mathbb{Q}^n \;\Leftrightarrow\; ||\bar{x}|| \leq x_1 \;\Leftrightarrow \left\{ \begin{array}{c} \bar{x}^T\bar{x} \leq x_1^2 \\ 0 \leq x_1 \end{array} \right\} \tag{2.26}$$

$$\Leftrightarrow \left\{ \begin{array}{c} \frac{\bar{x}^T\bar{x}}{x_1} \leq x_1 \\ 0 \leq x_1 \end{array} \right\} \Leftrightarrow \begin{pmatrix} x_1 & \bar{x}^T \\ \bar{x} & x_1 I_{n-1} \end{pmatrix} \succeq 0. \tag{2.27}$$

Where last equivalence is provided by Schur complement lemma (see appendix, Theorem A.1).

In case of more general standard form of SOCP (2.11), the $x \in \mathbb{Q}$ constraint can be transformed similarly.

$$x = (x^1, \dots, x^k)^T \Leftrightarrow M = diag(M_1, \dots, M_k) \succeq 0, \tag{2.28}$$

where M is a block diagonal matrix, with blocks M_i of the form (2.27), for $i = 1, \dots, k$, corresponding to the constraints $x^i \in \mathbb{Q}^{n_i}$.

2.4 Copositive and completely positive programming

The copositive programming (CP) and completely positive programming (CPP) are convex conic classes which are important mainly for theoretical interests since CCP and CP problems are numerically intractable. In fact, it is known that determining whether a matrix is copositive is co-NP-complete [26].

Even so, these optimization seem to play an important role in current research. Recently, multiple classes of QCQP have been equivalently reformulated as convex completely positive or copositive programs [20, 21, 22]. While the restating of a problem as an optimization problem over one of these cones does not resolve the difficulty of that problem, however, studying properties of $\mathbb{C}^n$ and $\mathbb{P}^n$ and using the conic formulations of quadratic and combinatorial problems does provide new insights and also computational improvements. (see the survey of Copositive programming [25])

Let us first define the copositive cone and completely positive cone.

Definition 2.29 (Copositive cone)**.** We say $\mathbb{C}^n$ is copositive cone of dimension n if

$$\mathbb{C}^n = \{M \in \mathbb{S}^n \mid x^T M x \geq 0 \ \forall x \in \mathbb{R}^n_+\}. \tag{2.30}$$

A matrix $M \in \mathbb{C}^n$ is called copositive.

Definition 2.31 (Completely positive cone)**.** We say $\mathbb{P}^n$ is completely positive cone of dimension n if

$$\mathbb{P}^n = \{M \in \mathbb{S}^n \mid M = \sum_{i=1}^{l} x^i x^{iT} \ \text{ where } x^i \in \mathbb{R}^n_+ \ (i = 1, \ldots, n)\}. \tag{2.32}$$

A matrix $M \in \mathbb{P}^n$ is called completely positive.

In fact, $\mathbb{C}^n$ and $\mathbb{P}^n$ are closed convex cones and they are dual cones to each other. That is $\mathbb{P}^n$ is the dual cone of the $\mathbb{C}^n$ and vice versa (see Examples A.11 , A.13 in the Appendix).

Definition 2.33. The primal–dual pair of the copositive (CP) and the completely positive (CPP) program is

$$\begin{array}{llcll} & \textit{Primal} & & & \textit{Dual} \\ \text{minimize} & A_0 \bullet X, & & \text{maximize} & b^T y, \\ \text{subject to} & A_k \bullet X = b_k, & & \text{subject to} & \sum_{k=1}^{m} y_k A_k + S = A_0, \\ & (k = 1, \ldots, m), & & & S \in \mathbb{P}^n, \\ & X \in \mathbb{C}^n, & & & \end{array} \tag{2.34}$$

where $X,\ S \in \mathbb{S}^n$ and $y \in \mathbb{R}^m$ are the variables; matrices $A_0, \ldots, A_m \in \mathbb{S}^n$ and vector $b = (b_1, \ldots, b_m)^T \in \mathbb{R}^m$ are given problem data.

Similarly to SDP (2.25), this problem can be reformulated using $svec(\cdot)$ operator to obtain the same form as all the previous classes with vector variable and standard inner product as an objective.

2.4.1 Relation to previous classes

In this section we will first describe the relations of $\mathbb{C}^n$ and $\mathbb{P}^n$ to the other cones. Then we will show that SDP is a subclass of CP, and in the end we will provide a simple example of the equivalent CPP reformulation for a special class of nonconvex QP.

Relation to other cones

First, let us define another two important cones

Definition 2.35. We will denote

$$\begin{array}{rcll} \Gamma & := & \{xx^T \mid x \in \mathbb{R}^n_+\} & \text{the cone of positive rank 1 matrices,} \\ \mathbb{N}^n & := & \{M \in \mathbb{S}^n \mid M_{ij} \geq 0, \forall\ 1 \leq i, j \leq n\} & \text{the nonnegative cone.} \end{array}$$

and say a matrix $M \in \mathbb{N}^n$ is nonnegative. Especially, when $M \in \mathbb{S}^n \cap \mathbb{N}^n$ then we call M a doubly nonnegative matrix, and we will refer to $\mathbb{S}^n \cap \mathbb{N}^n$ as doubly nonnegative cone.

A following proposition states the relations between positive semidefinite, copositive and completely positive matrices.

Proposition 2.36. *It holds that*

$$\Gamma \subset \mathbb{P}^n \subset \mathbb{S}^n \cap \mathbb{N}^n \subset \mathbb{S}^n \subset \mathbb{S}^n + \mathbb{N}^n \subset \mathbb{C}^n.$$

Lemma 2.37. *It holds that* $\mathbb{P}^n \subset \mathbb{S}^n_+ \subset \mathbb{C}^n$.

Proof. For a completely positive matrix $M = \sum_i x^i (x^i)^T$ with $x^i \in \mathbb{R}^n_+$ and an arbitrary vector $y \in \mathbb{R}^n_+$ it holds that

$$y^T M y = \sum_i y^T x^i (x^i)^T y = \sum_i (y^T x^i)^2 \geq 0.$$

Therefore M is positive semidefinite and since this is true for any $M \in \mathbb{P}^n$ we have the first inclusion $\mathbb{P}^n \subset \mathbb{S}^n_+$.

A positive semidefinite matrix $M \in \mathbb{S}^n_+$ suffices $x^T M x \geq 0$ for all $x \in \mathbb{R}^n$, therefore also for all $x \in \mathbb{R}^n_+$. Hence $\mathbb{S}^n_+ \subset \mathbb{C}^n$. □

Proof of Proposition 2.36. We will proceed through inclusions from left to right. The first inclusion $\Gamma \subset \mathbb{P}^n$ is trivial. It is also easy to see that all matrices in $\mathbb{P}^n$ are nonnegative, therefore $\mathbb{P}^n \subset \mathbb{N}^n$. Together with $\mathbb{P}^n \subset \mathbb{S}^n_+$ from th lemma we have $\mathbb{P}^n \subset \mathbb{S}^n \cap \mathbb{N}^n$. The assertions $\mathbb{S}^n \cap \mathbb{N}^n \subset \mathbb{S}^n \subset \mathbb{S}^n + \mathbb{N}^n$ are trivial. Now we only need to prove that $\mathbb{S}^n + \mathbb{N}^n \subset \mathbb{C}^n$. But this is an easy exercise since $\mathbb{C}^n$ contains both $\mathbb{S}^n$ and $\mathbb{N}^n$ (former by lemma, latter trivially), and $\mathbb{C}^n$ is convex (see Example A.11 in the Appendix). □

Relation to SDP

To establish that SDP is a subclass of copositive programming we only need to reformulate the semidefinite constraint $X \succeq 0$ in terms of linear and copositive cone constraints.

Every vector $v \in \mathbb{R}^n$ can be written as difference of two positive vectors $v = x - y$, $x, y \in \mathbb{R}^n_+$. Therefore a symmetric matrix $X \in \mathbb{S}^n$ is positive semidefinite if and only if

$$(x-y)^T X (x-y) \geq 0, \quad \forall x, y \in \mathbb{R}^n_+$$

This condition can be rewritten as

$$u^T C u \geq 0, \ \forall u \in \mathbb{R}^n_+, \quad \text{where} \quad u = \begin{pmatrix} x \\ y \end{pmatrix}, \quad \text{and} \quad C = \begin{pmatrix} X & -X \\ -X & X \end{pmatrix}.$$

Which is equivalent to $C \in \mathbb{C}^{2n}$. Also it is clear that this form of C can be enforced with linear equalities, therefore semidefinite programming is indeed a subclass of copositive programming.

Relation to QP

In 2009, completely positive programming (CPP) relaxation for a class of binary QOPs was proposed by Burer [21]. The class was extended to a more general class of QOPs by Eichfelder and Povh [22] and by Arima, Kim and Kojima [20]. Theoretically strong results were presented in their papers [20, 21, 22]

Copositive programming is closely related to quadratic and combinatorial optimization. We illustrate this connection by means of the standard quadratic problem (an example from [25]).

$$\begin{array}{ll} \text{minimize} & x^T Q x, \\ \text{subject to} & e^T x = 1, \\ & x \in \mathbb{R}^n_+, \end{array} \tag{2.38}$$

where e denotes the all ones vector. Rewriting the objective $x^T Q x = Q \bullet x x^T$ and linear constraint $e^T x = 1$ as $ee^T \bullet xx^T = 1$, one can see that

$$\begin{array}{ll} \text{minimize} & Q \bullet X, \\ \text{subject to} & ee^T \bullet X = 1, \\ & X \in \mathbb{P}^n, \end{array} \tag{2.39}$$

is a relaxation of (2.38). Since the objective is now linear, an optimal solution must be attained in an extremal point of the convex feasible set. It can be shown that these extremal points are exactly the positive rank 1 matrices xx^T with $x \in \mathbb{R}^n_+$ and $e^T x = 1$ (see the Lemma 2.40 bellow). Together, these results imply that (2.39) is in fact an exact reformulation of (2.38).

Lemma 2.40. *It holds that*

$$\{X \in \mathbb{P}^n \mid ee^T \bullet X = 1\} = conv\left(\{xx^T \mid x \in \mathbb{R}^n_+,\ e^T x = 1\}\right).$$

Proof. Let $X \in \mathbb{P}^n$ satisfy $ee^T \bullet X = 1$. We will show that X can be written as convex combination of vectors from $\Gamma_1 := \{xx^T \mid x \in \mathbb{R}^n_+,\ e^T x = 1\}$.

From the definition of $\mathbb{P}^n$, there are $x^1, \ldots, x^l \in \mathbb{R}^n_+$, such that

$$X \sum_{i=1}^{l} x^i (x^i)^T = \sum_{i=1}^{l} (e^T x^i)^2 \left(\frac{x^i}{e^T x^i}\right) \left(\frac{x^i}{e^T x^i}\right)^T = \sum_{i=1}^{l} \alpha_i y^i (y^i)^T$$

where $\alpha_i = (e^T x^i)^2$ and $y^i = x^i/(e^T x^i)$. It is easy to see that $y^i \in \mathbb{R}^n_+$ and $e^T y^i = 1$, so $y^i (y^i)^T \in \Gamma_1$. Also, it holds that $\sum \alpha_i = 1$, since

$$1 = ee^T \bullet X = ee^T \bullet \sum_{i=1}^{l} x^i (x^i)^T = \sum_{i=1}^{l} ee^T \bullet x^i (x^i)^T = \sum_{i=1}^{l} (e^T x^i)^2 = \sum_{i=1}^{l} \alpha_i.$$

Therefore $X = \sum_{i=1}^{l} \alpha_i y^i (y^i)^T$ is a convex combination of vectors from Γ_1, hence $X \in conv(\Gamma_1)$.

□

2.5 Conic programming

All of the previous mentioned classes are quite similar. With respect to their variable space, all of them have linear objective, linear constraints and cone constraint.

In fact, they are special cases of the so called conic linear programs.

Definition 2.41 (Conic Programming)**.** The primal–dual pair of the Linear Conic Program in the standard form is

$$\begin{array}{llcll} & \textit{Primal} & & & \textit{Dual} \\ \text{minimize} & c^T x, & & \text{maximize} & b^T y, \\ \text{subject to} & Ax = b, & & \text{subject to} & A^T y + s = c, \\ & x \in \mathcal{K}, & & & s \in \mathcal{K}^*, \end{array} \qquad \text{(Conic Program)}$$

where $x \in \mathbb{R}^n$, $y \in \mathbb{R}^m$ and $s \in \mathbb{R}^n$ are the variables; and $m \times n$ real matrix A, vectors $c \in \mathbb{R}^n$, $b \in \mathbb{R}^m$, $c \in \mathbb{R}^n$, and the proper cone $\mathcal{K}$ are given problem data. The

$$K^* = \{z \mid \forall x \in \mathcal{K},\ x^T z \geq 0\}, \tag{2.42}$$

denotes the dual cone of $\mathcal{K}$ (see The Section about Cones in Appendix A.2).

Conic programming contains, but is not limited to, any problems combined from LP, SOCP and SDP programs. For example

$$\begin{array}{ll} \text{minimize} & c^T x, \\ \text{subject to} & A^i x^i = b^i,\ (i = 1, \ldots, k), \\ & x = (x^1, \ldots, x^k) \in \mathcal{K} = (\mathcal{K}^1, \ldots, \mathcal{K}^k), \end{array} \tag{2.43}$$

where the variable $x = (x^1, \ldots, x^k)^T$ is the Cartesian product of the variables x^i, constrained by various LP, SOCP or SDP constraints $A^i x^i = b$, $x_i \in \mathcal{K}^i$, where each $\mathcal{K}^i$ is either nonegative orthant, second order cone or semidefinite cone.

This is due to the fact, that all cones we have talked about so far are proper cones (i.e. nonegative orthant $\mathbb{R}^n_+$, second order cone $\mathbb{Q}^n$ and semidefinite cone $\mathbb{S}^n_+$ as a subset of $\mathbb{R}^{n(n+1)/2}$). For proper cones it holds that their Cartesian product is again proper cone (see Proposition A.10 and Example A.12 in the Appendix).

2.6 Dual problems

2.6.1 Dual problem of conic programming

We will derive the dual forms of LP, SOCP, SDP all at once by deriving Lagrange dual of general conic program

$$\begin{array}{ll} \text{minimize} & c^T x, \\ \text{subject to} & Ax = b, \\ & x \in \mathcal{K}. \end{array} \tag{2.44}$$

The Lagrangian of the problem is given by $\mathcal{L} : \mathbb{R}^n \times \mathbb{R}^m \times \mathcal{K}^* \to \mathbb{R}$,

$$\mathcal{L}(x, y, s) = c^T x + y^T(b - Ax) - s^T x. \tag{2.45}$$

The last term (notice that $s \in \mathcal{K}^*$) is added to take account of the conic constraint $x \in \mathcal{K}$. It is with negative sign in order to have $\mathcal{L}(x, \cdot, \cdot) \le c^T x$ for all x feasible in (2.44). Indeed, from the very definition of dual cone:

$$\sup_{s \in \mathcal{K}^*} -s^T x = \begin{cases} 0 & \text{if } x \in \mathcal{K}, \\ +\infty & \text{otherwise.} \end{cases} \tag{2.46}$$

Therefore, the Lagrange dual function is

$$\begin{aligned} g(y, s) &= \inf_x \; \mathcal{L}(x, y, s) &(2.47)\\ &= \inf_x \; y^T b + (c + A^T y - s)^T x &(2.48)\\ &= \begin{cases} b^T y & \text{if } c - A^T y - s = 0, \\ -\infty & \text{otherwise.} \end{cases} &(2.49) \end{aligned}$$

Hence, the dual problem of linear conic programming in the standard form is

$$\begin{array}{ll} \text{maximize} & b^T y \\ \text{subject to} & A^T y + s = c \\ & s \in \mathcal{K}^* \end{array} \tag{2.50}$$

Since $\mathbb{R}^n_+$, $\mathbb{Q}^n$ and $\mathbb{S}^n_+$ are self-dual, by replacing the $\mathcal{K}$ (and $\mathcal{K}^*$) with any of these cones, we get the dual of standard LP, SOCP and SDP as given in the Section 2.

Remark 2.51. For a closed cone $\mathcal{K}$ it holds that $\mathcal{K}^{**} = \mathcal{K}$ (where $\mathcal{K}^{**}$ denotes dual cone of the dual cone). Therefore, repeating the above procedure, one can easily show that dual of this dual problem is the original primal program (2.44).

2.6.2 Dual problem of QCQP

We will derive dual form of standard QCQP

$$\begin{array}{ll} \text{minimize} & x^T P_0 x + q_0^T x + r_0 \\ \text{subject to} & x^T P_k x + q_k^T x + r_k \le 0, \;\; (k = 1, \dots, m) \end{array} \tag{2.52}$$

The Lagrangian of the problem is given by $\mathcal{L} : \mathbb{R}^n \times \mathbb{R}^m_+ \to \mathbb{R}$,

$$\begin{aligned} \mathcal{L}(x, y) &= x^T P_0 x + q_0^T x + r_0 + \sum_{k=1}^m y_k (x^T P_k x + q_k^T x + r_k) &(2.53)\\ &= x^T P(y) x + q(y)^T x + r(y), &(2.54) \end{aligned}$$

where

$$P(y) = P_0 + \sum_{k=1}^m y_k P_k, \quad q(y) = q_0 + \sum_{k=1}^m y_k q_k, \quad r(y) = r_0 + \sum_{k=1}^m y_k r_k. \tag{2.55}$$

It holds that $\inf_x \mathcal{L}(x,y) > -\infty$ if and only if $P(y) \succeq 0$ and there exists $\hat{x}$ such that $P(y)\hat{x} + q(y) = 0$.

Thus, the Lagrange dual function is

$$\begin{aligned} g(y) &= \min_x \ \mathcal{L}(x,y) && (2.56)\\ &= \begin{cases} -\frac{1}{4}q(y)^T P(y)^\dagger q(y) + r(y) & \text{if } P(y) \succeq 0, \quad q(y) \in \mathcal{R}(P(y)) \\ -\infty & \text{otherwise,} \end{cases} && (2.57) \end{aligned}$$

where $P^\dagger$ denotes Moore-Penrose pseudoinverse of P (see appendix). Finally, dual form of standard QCQP problem is

$$\begin{array}{ll} \text{maximize} & -\frac{1}{4}q(y)^T P(y)^\dagger q(y) + r(y), \\ \text{subject to} & y \geq 0, \\ & P(y) \succeq 0, \\ & \mathcal{R}(q(y)) \subseteq \mathcal{R}(P(y)), \end{array} \qquad \text{(QCQP Dual)}$$

where $y \in R^m$ is variable; and problem data $P_0, P_1 \dots, P_m, q_0, q_1 \dots, q_m, r_0, r_1 \dots, r_m$ are given from the primal QCQP above.

This dual problem is basically a SDP (in the LMI form). We first rewrite the objective as linear function t with additional constraint.

$$\begin{array}{ll} \text{maximize} & t, \\ \text{subject to} & t \leq -\frac{1}{4}q(y)^T P(y)^\dagger q(y) + r(y), \\ & y \geq 0, \\ & P(y) \succeq 0, \\ & \mathcal{R}(q(y)) \subseteq \mathcal{R}(P(y)). \end{array} \qquad (2.58)$$

Due to the Schur complement lemma (see appendix, Theorem A.1) the above is equivalent to

$$\begin{array}{ll} \text{maximize} & t, \\ \text{subject to} & M := \begin{pmatrix} r(y) - t & \frac{1}{2}q(y)^T \\ \frac{1}{2}q(y) & P(y) \end{pmatrix} \succeq 0, \\ & y \geq 0, \end{array} \qquad (2.59)$$

where the matrix M is easily expanded as

$$M = M_0 + \sum_{k=1}^{m} y_k M_k - tE \qquad (2.60)$$

with

$$M_0 = \begin{pmatrix} r_0 & \frac{1}{2}q_0^T \\ \frac{1}{2}q_0 & P_0 \end{pmatrix}, \quad M_k = \begin{pmatrix} r_k & \frac{1}{2}q_k^T \\ \frac{1}{2}q_k & P_k \end{pmatrix}, \quad E = \begin{pmatrix} 1 & 0_n^T \\ 0_n & 0_{n\times n} \end{pmatrix}. \qquad (2.61)$$

2.6.3 Second dual of QCQP

Now the whole process can be repeated. Create Lagrangian $\mathcal{L}_d : \mathbb{R} \times \mathbb{R}_+^m \times \mathbb{S}_+^{n+1} \times \mathbb{R}_+^m \to \mathbb{R}$,

$$\mathcal{L}_d(t,y,Y,u) = t + Y \bullet M_0 + \sum_{k=1}^{m} y_k Y \bullet M_k - tY \bullet E + u^T y, \qquad (2.62)$$

where

$$Y = \begin{pmatrix} x_0 & x^T \\ x & X \end{pmatrix} \succeq 0. \tag{2.63}$$

The Lagrange dual function is

$$\begin{aligned} g_d(Y, u) &= \sup_{t,y} \mathcal{L}_d(t, y, Y, u) && (2.64) \\ &= \begin{cases} Y \bullet M_0, & \text{if } \left[\begin{array}{l} Y \bullet M_k \leq 0, \ k = 1, \dots, m, \\ x_0 = 1 \ \text{ and } \ u = 0_m, \end{array} \right. \\ \infty, & \text{otherwise.} \end{cases} && (2.65) \end{aligned}$$

It is easy to see that with $x_0 = 1$ and Schur complement lemma we have

$$\begin{aligned} Y \bullet M_0 &= P_0 \bullet X + q_0^T x + r_0, && (2.66) \\ Y \bullet M_k \leq 0 &\Leftrightarrow P_k \bullet X + q_k^T x + r_k \leq 0, && (2.67) \\ Y \succeq 0 &\Leftrightarrow X \succeq xx^T. && (2.68) \end{aligned}$$

Thus we obtain the following SDP as second dual of QCQP

$$\begin{array}{ll} \text{minimize} & P_0 \bullet X + q_0^T x + r_0 \\ \text{subject to} & P_k \bullet X + q_k^T x + r_k \leq 0, \ (k = 1, \dots, m) \\ & X \succeq xx^T. \end{array} \tag{2.69}$$

Chapter 3

Relaxations

It was mentioned in the beginning that our strategy is to relax the nonconvex QCQPs (1.2) to easier problem. Let us first explore what is a relaxation and how can it be useful.

Relaxation is usually freely understood as an optimization problem which is obtained by relaxing (loosening) some constraints or even by approximating objective function with a different one. The goal is to obtain a problem which is easier to solve, but still carries some kind of information about the original one. For example, solving the relaxation may give an approximation of the original problem solution.

Although relaxation is a common term in optimization, we did not manage to find any rigorous definition of this notion. At least for the needs of this thesis we would like to propose a definition covering everything we refer to as relaxation.

Definition 3.1 (Proposed definition)**.** Let P, Q be a minimization problems

$$\begin{array}{llcll} & P & & & Q \\ \text{minimize} & f(x), & & \text{minimize} & g(y), \\ \text{subject to} & x \in X, & & \text{subject to} & y \in Y, \end{array} \tag{3.2}$$

Where $f: X \to \mathbb{R}$, $g: Y \to \mathbb{R}$ are objective functions and X, Y are arbitrary closed sets (for the minimum to exist). We say that Q is a relaxation of P if there is and an injective mapping $u: X \to Y$. such that $g(u(x)) \leq f(x)$ for all $x \in X$.

Remark 3.3. The existence of such injective mapping u is equivalent to the existence of a a set $Y' \subseteq Y$ and a surjective mapping $v: X \to Y'$ such that $g(y') \leq f(v(y'))$ for all $y' \in Y'$. Therefore it is equivalent to state that Q is a relaxation of P if there is $Y' \subset Y$ and a surjective mapping $v: Y' \to X$, such that $g(y') \leq f(v(y'))$ for all $y' \in Y'$.

For example, any problem Q with the same objective $g = f$ and extended feasible set $Y \supseteq X$ is a relaxation of P (in that case $Y' = X$ and $v(x) = x$ suit the definition).

The plus of this definition is that problems P and Q may have different variable spaces. This is quite common, such a relaxation is for example semidefinite programming relaxation, where original QCQP (1.2) with variable $x \in \mathbb{R}^n$ is relaxed as

SDP (3.12) with variable $Y \in \mathbb{S}^{n+1}$. In this particular relaxation the mapping v is obvious from the construction of Y as

$$Y = \begin{pmatrix} 1 & x^T \\ x & X \end{pmatrix}.$$

An interesting point of view is that the relaxation approximates the problem from the "outside", giving a lower bound on the optimal value for the original problem. The other option is to approximate problem form the "inside" for example by restricting the feasible set or penalizing the objective, resulting to an upper bound on the objective of the original problem.

One could say, that relaxation of minimization problem

$$\begin{array}{ll} \text{minimize} & f(x), \\ \text{subject to} & x \in X, \end{array} \tag{3.4}$$

is another minimization problem

$$\begin{array}{ll} \text{minimize} & f_R(x), \\ \text{subject to} & x \in X_R, \end{array} \tag{3.5}$$

with properties $X \subseteq X_R$ and $c_R(x) \leq c(x) \forall x \in X$. It easily follows, that solving the relaxed problem will provide a lower bound on the optimal value of original problem. In some cases we can also extract a feasible solution of the original problem from solution of the relaxation. In that case we obtain an upper bound for the optimal value.

Moreover, these bounds may not only give us an idea about the optimal value, but also, may provide means to find an optimal solution of the original problem.

In the following we will explore the well known SDP relaxation which is basically casting the nonconvex QCQP into the convex SDP class. Furthermore, we will introduce some of the approaches for either loosening the SDP relaxation (in order to gain more speed) or strengthening these resulting SDP, SOCP, convex QP and LP relaxations (in order to obtain tighter bounds).

3.1 SDP relaxation of QCQP

Since Goemans and Williamson [37] proposed the SDP relaxation of the max-cut problem and proved its 0.878 approximation bound of the optimal value, a lot of work have been focused on solving the nonconvex (mostly combinatorial) QP problems using SDP relaxation methods. In this section we will derive the standard SDP relaxation of QCQP.

Consider the QCQP (1.2). Using identity $x^T P_k x = P_k \bullet xx^T$, which follows from the Definiton 2.18, it can be rewritten as follows

$$\begin{array}{ll} \text{minimize} & P_0 \bullet X + q_0^T x + r_0, \\ \text{subject to} & P_k \bullet X + q_k^T x + r_k \leq 0, \; (k = 1, \dots, m) \\ & X = xx^T. \end{array} \tag{3.6}$$

Notice, that the variable X is a symmetric $n \times n$ matrix. The problem can be reformulated in the following way:

$$\begin{array}{ll} \text{minimize} & M_0 \bullet Y, \\ \text{subject to} & M_k \bullet Y \leq 0, \; (k = 1, \dots, m), \\ & X = xx^T, \end{array} \tag{3.7}$$

where ,

$$M_k = \begin{pmatrix} \alpha_k & \frac{1}{2} q_k^T \\ \frac{1}{2} q_k & P_k \end{pmatrix}, \qquad Y = \begin{pmatrix} 1 & x^T \\ x & X \end{pmatrix}. \tag{3.8}$$

This problem still has a non-convex constraint $X = xx^T$, which can be relaxed by a convex constraint, as stated in the following lemma.

Lemma 3.9. *Let $x \in \mathbb{R}^n$, an $n \times n$ symmetric matrix X, and $n+1 \times n+1$ symmetric matrix Y such that*

$$Y = \begin{pmatrix} 1 & x^T \\ x & X \end{pmatrix}.$$

Then

(i) $X \succeq xx^T$ if and only if $Y \succeq 0$.

(ii) $X = xx^T$ holds if and only if $Y \succeq 0$ and $\operatorname{rank} Y = 1$.

Proof. (i) The statement follows from Schur complement lemma for PSD (see Theorem A.1 in the appendix)

(ii) ($\Rightarrow$) If $X = xx^T$, then also $X \succeq xx^T$, thus $Y \succeq 0$ holds by (i). And since $X = xx^T$,

$$Y = \begin{pmatrix} 1 & x^T \\ x & xx^T \end{pmatrix} = \begin{pmatrix} 1 \\ x \end{pmatrix} (1, \; x^T).$$

Hence $\operatorname{rank} Y = 1$.
($\Leftarrow$) Let $Y \succeq 0$ and $\operatorname{rank} Y = 1$. Since $\operatorname{rank} Y = 1$, each row of Y must be scalar multiple of the first (obviously nonzero) row $(1, \; x^T)$. To match the first column the $(i+1)$-st row of Y must be $x_i(1, \; x^T)$, for $i = 1, \dots, n$. Therefore, $X = xx^T$. □

Remark 3.10. Notice that we have proven last implication of (ii) without using $Y \succeq 0$. In fact it is redundant. It also holds that $X = xx^T \Leftrightarrow \operatorname{rank} Y = 1$. In fact, there are only 2 options for Y of rank 1: $Y = vv^T$ or $Y = -vv^T$. The second option is easily excluded, because $Y_{11} = 1 > 0$. However, this redundant constraint $Y \succeq 0 \Leftrightarrow X \succeq xx^T$ will let us keep something from the rank 1 constraint after relaxing it. This approach of adding the redundant constraints (also known as valid inequalities) is often usefull for strengthening the relaxation. For more on valid inequalities see [12, 13, 14].

Using the Lemma 3.9 we obtain another equivalent formulation of the original QCQP

$$\begin{array}{ll} \text{minimize} & M_0 \bullet Y, \\ \text{subject to} & M_k \bullet Y \leq 0, \ (k = 1, \ldots, m) \\ & Y \succeq 0, \quad \text{rank } Y = 1. \end{array} \tag{3.11}$$

relaxing the nonconvex rank 1 constrain we get the following SDP relaxation of (1.2)

$$\begin{array}{ll} \text{minimize} & M_0 \bullet Y, \\ \text{subject to} & M_k \bullet Y \leq 0, \ (k = 1, \ldots, m) \\ & Y \succeq 0. \end{array} \tag{3.12}$$

Expanding the terms $M_k \bullet Y$ we obtain the standard SDP relaxation

$$\begin{array}{ll} \text{minimize} & P_0 \bullet X + q_0^T x + r_0, \\ \text{subject to} & P_k \bullet X + q_k^T x + r_k \leq 0, \ (k = 1, \ldots, m) \\ & X \succeq xx^T. \end{array} \tag{3.13}$$

Remark 3.14. Notice that this is exactly the second dual of QCQP (2.69).

The above relaxed problem has different variable space $(X, x) \in \mathbb{R}^{n \times n} \times \mathbb{R}^n$ than original problem $x \in \mathbb{R}^n$. In other words, the variable space increased from $O(n)$ to $O(n^2)$ variables.

3.2 Convex QP and SOCP relaxation

??

These relaxations are expected to provide weaker bounds in less time compared to SDP relaxations. In fact, SOCP relaxations are often constructed as further relaxations of SDP relaxations. So, in certain sense, one can see that SOCP relaxations are never tighter than their SDP counterparts. [1]

Using the procedure from section 2.2, any convex instance of QCQP can be formulated as SOCP. Therefore we can consider any convex quadratic relaxation as SOCP relaxation. Specifically, a convex QCQP relaxation may be represented as

$$\begin{array}{ll} \text{minimize} & x^T B_0 x + b_0^T x + \beta_0 \\ \text{subject to} & x^T B_k x + b_k^T x + \beta_i \leq 0, \ (k = 1, \ldots, l) \end{array} \tag{3.15}$$

where all $B_k \succeq 0$ for $k = 0, \ldots, l$. We say that (3.15) is SOCP relaxation of QCQP (1.2) if any x feasible in (1.2) is also feasible in (3.15) and $x^T B_0 x + b_0^T x + \beta_0 \leq x^T P_0 x + q_0^T x + r_0$ holds.

A possible approach to forming such a SOCP (or convex QP) relaxation is by further relaxing the SDP relaxation

$$\begin{array}{ll} \text{minimize} & P_0 \bullet X + q_0^T x + r_0, \\ \text{subject to} & P_k \bullet X + q_k^T x + r_k \leq 0, \ (k = 1, \ldots, m) \\ & X \succeq xx^T. \end{array} \tag{3.16}$$

The reason why this is even considered is that solving SDP is computationally costly operation and even if it offers better bounds it may not be the best option. For example in the branch and bound procedure one needs to solve a large sequence of relaxations in order to obtain the optimal solution (see the Section 5.1 about branch and bound).

In this section we will introduce strategies for further loosening of the SDP relaxation. There are two main reasons why SDP is expensive to solve, the $O(n^2)$ variable space and semidefinite constraint. We will first address each of them separately and then provide an example of combining both of the approaches

3.2.1 SOCP relaxation of semidefinite constraint

Semidefinite constraint $X - xx^T \succeq 0$ in (3.16) is equivalent to $C \bullet (X - xx^T) \geq 0$ for all $C \in S^n_+$. Using this fact, authors of [4] propose SOCP relaxation of the semidefinite constraint $X - xx^T \succeq 0$ by replacing it with multiple constraints of the form

$$x^T C_i x - C_i \bullet X \leq 0 \quad (i = 1, \dots, t).$$

Since $C_i \succeq 0$, these are convex quadratic constraints and using the procedure described earlier (in the Section 2.2.1) one can formulate them equivalently as a second order cone constraints of the form

$$\begin{pmatrix} v_0^i \\ v^i \end{pmatrix} = \begin{pmatrix} 1 + C_i \bullet X \\ 1 - C_i \bullet X \\ 2L_i^T x \end{pmatrix}, \quad ||v^i|| \leq v_0^i, \quad (i = 1, \dots, l), \tag{3.17}$$

where L_i is obtained from the Choelsky decomposition of $C_i = L_i L_i^T$.

They also show how to extract these convex inequalities from original quadratic inequality constraints. We will omit indices and consider the constraint $x^T P x + q^T x + r \leq 0$. Let

$$\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_l \geq 0 > \lambda_{l+1} \geq \dots \geq \lambda_n, \tag{3.18}$$

be the eigenvalues of the matrix P and let $u_1, \dots, u_n$ be the corresponding eigenvectors, such that

$$\|u_i\|_2 = 1 \ (i = 1, \dots, n) \quad \text{and} \quad u_i^T u_j = 0 \ (i \neq j). \tag{3.19}$$

Then $P = \sum_{j=1}^{h} \lambda_j u_j u_j^T$, and any matrix of the either form

$$C^+ \ = \ \sum_{j \in J} \lambda_j u_j u_j^T \quad \text{for } J \subseteq \{1, \dots, l\}, \tag{3.20}$$

$$C^- \ = - \sum_{j \in J} \lambda_j u_j u_j^T \quad \text{for } J \subseteq \{l+1, \dots, n\}, \tag{3.21}$$

is positive semidefinite and can be used in $x^T C x - C \bullet X \leq 0$ constraint.

The cost of solving the resulting SOCP depends very much on the ranks of $C_i \in \mathbb{S}^n_+$, the larger their ranks are, the more auxiliary variables we need to introduce and the more expensive the cost of solving the resulting SOCP becomes. In an attempt to keep the amount of computation small, low rank C_i are reasonable. Authors of [4] suggest putting $C_i = e_i e_i^T$, which corresponds to the constraints

$$x_i^2 - X_{ii} \leq 0. \tag{3.22}$$

They also employ rank-1 convex quadratic inequalities tied with the problem data, taking $C_i = u_i u_i^T$. where u_i are chosen as eigenvectors of the matrices P_k from the quadratic inequality constraints of original QCQP. This choice corresponds to constraints

$$x^T u_i u_i^T x - u_i u_i^T \bullet X \leq 0 \ (i = 1, \ldots, l), \tag{3.23}$$

3.2.2 SOCP in the original variable

The SOCP relaxation introduced above has avoided the semidefinite constraint but still includes a matrix variable X from the SDP which increases size of the problem dramatically (from n to $n + n(n+1)/2$). In [4] authors also introduce a general relaxation scheme to obtain SOCP relaxation of QCQP (1.2) in the original variable space $x \in \mathbb{R}^n$. First they assume that objective function is linear (otherwise we can add new variable $t \geq x^T P_0 x + q_0^T + r_0$ and then minimize t). Then each P_k is written as

$$P_k = P_k^+ - P_k^-, \quad \text{where} \quad P_k^+, P_k^- \succeq 0, \quad k = 1, \ldots, m.$$

So that each constraint can be expressed as

$$x^T P_k^+ x + q_k^T x + r_k \leq x^T P_k^- x. \tag{3.24}$$

Then an auxiliary variable $z_k \in \mathbb{R}$ is introduced to represent $x^T P_k^- x = z_k$, but also immediately relaxed as $x^T P_k^- x \leq z_k$, resulting in covnex system

$$\begin{array}{rl} x^T P_k^+ x + q_k^T x + r_k & \leq z_k \\ x^T P_k^- x & \leq z_k. \end{array} \tag{3.25}$$

Finally, z_k must be bounded in some fashion, say as $z_k \leq \mu \in \mathbb{R}$, or else the relaxation would be useless. In the next section we will show an example of such bounds for z_k. In this way convex QCQP relaxation is constructed and it is simply transformed to SOCP using the procedure from [4] described in the Section 2.2.

3.2.3 Relaxing the semidefinite constraint and reducing the number of variables

The idea of combining both above approaches (introduced in [4]) is to relax each quadratic inequality constraint together with constraints of the form $x^T Cx - C \bullet X \leq 0$ relaxing the semidefinite constraint. We will omit the indices and consider the constraint

$$x^T Px + q^T x + r \leq 0. \tag{3.26}$$

Let the eigenvalues λ_i and eigenvectors u_i of P for $(i = 1, \ldots, n)$ be as above, see (3.18) and (3.19). And let $P = P^+ - P^-$ with

$$P^+ = \sum_{j=1}^{l} \lambda_j u_j u_j^T \succeq 0 \quad , \quad P^- = -\sum_{j=l+1}^{n} \lambda_j u_j u_j^T \succeq 0, \tag{3.27}$$

then we may rewrite the quadratic inequality constraint $x^T P x + q^T x + r \leq 0$ as

$$\begin{array}{l} x^T P^+ x + \sum_{j=l+1}^{n} \lambda_j z_j + q^T x + r \leq 0, \\ x^T (u_j u_j^T) x - z_j = 0, \ (j = l+1, \ldots, n., \end{array} \tag{3.28}$$

Now, relaxing the last $n - l$ inequalities we obtain a set of convex inequalities

$$\begin{array}{l} x^T P^+ x + \sum_{j=l+1}^{n} \lambda_j z_j + q^T x + r \leq 0 \\ x^T (u_j u_j^T) x - z_j \leq 0, \ (j = l+1, \ldots, n) \end{array} \tag{3.29}$$

It is necessary to add the appropriate constraints on the variables z_j to bound them from above, authors show that $\sum_{j=l+1}^{n} z_j \leq \|x\|_2^2$ follows from (3.28). In fact, if x and z_j for $(j = l+1, \ldots, n)$ satisfy (3.28) then

$$\sum_{j=l+1}^{n} z_j = \sum_{j=l+1}^{n} x^T (u_j u_j^T) x \leq x^T \left(\sum_{j=1}^{n} u_j u_j^T \right) x \leq \|x\|_2^2, \tag{3.30}$$

where the last inequality holds because $\left(\sum_{j=1}^{n} u_j u_j^T \right)$ is an orthonormal matrix.

So the final relaxation of the $x^T P x + q^T x + r \leq 0$ constraint is

$$\begin{array}{l} x^T P^+ x + \sum_{j=l+1}^{n} \lambda_j z_j + q^T x + r \leq 0 \\ x^T (u_j u_j^T) x - z_j \leq 0, \ (j = l+1, \ldots, n) \\ \sum_{j=l+1}^{n} z_j \leq \|x\|_2^2 \end{array} \tag{3.31}$$

In the end authors of [4] compare this relaxation and the one obtained by relaxing the semidefinite constraint without reducing the number of variables. Consider the following relaxation of the same quadratic constraint (3.26) together with valid inequalities relaxing the semidefinite constraint

$$\begin{array}{l} P \bullet X + q^T x + r \leq 0, \\ x^T P^+ x - P^+ \bullet X \leq 0, \\ x^T u_j u_j^T x - u_j u_j^T \bullet X \leq 0, \ (j = l+1, \ldots, n). \end{array} \tag{3.32}$$

Suppose $(x, X) \in \mathbb{R}^n \times \mathbb{S}^n$ satisfy this relaxation, then x and $z_j = u_j^T X u_j$ for $j = l+1, \ldots, n$ satisfy (3.31). So we may see (3.31) as further relaxation of (3.32). Therefore it will provide weaker bounds, however it will take less time to solve because of the smaller variable space. For further details about this approach see [3, 4].

3.3 Mixed SOCP-SDP relaxation

In [1] authors Burer, Kim and Kojima have introduced compromise, relaxation of the QCQP (1.2) somewhere between SDP and SOCP. We will describe their approach with the special case they provide as introduction. We are dealing with QCQP (1.2)

$$\begin{array}{ll} \text{minimize} & x^T P_0 x + q_0^T x + r_0 \\ \text{subject to} & x^T P_k x + q_k^T x + r_k \leq 0, \ (k = 1, \dots, m). \end{array} \tag{3.33}$$

Let $\lambda_{min}(P_k)$ denote smallest eigenvalue of P_k. For all $k = 0, \dots, m$ define $\lambda_k = -\lambda_{min}(P_k)$ so that $P_k + I\lambda_k \succeq 0$. Then (1.2) is equivalent to

$$\begin{array}{ll} \text{minimize} & -\lambda_0 x^T x + x^T(P_0 + \lambda_0 I)x + q_0^T x + r_0 \\ \text{subject to} & -\lambda_k x^T x + x^T(P_k + \lambda_k I)x + q_k^T x + r_k \leq 0, \ (k = 1, \dots, m) \end{array} \tag{3.34}$$

which has following SOCP-SDP relaxation

$$\begin{array}{ll} \text{minimize} & -\lambda_0 Tr(X) + x^T(P_0 + \lambda_0 I)x + q_0^T x + r_0 \\ \text{subject to} & -\lambda_k Tr(X) + x^T(P_k + \lambda_k I)x + q_k^T x + r_k \leq 0, \\ & (k = 1, \dots, m) \\ & X \succeq xx^T \end{array} \tag{3.35}$$

Notice that other than $X \succeq xx^T$, the only variables in X to appear in the program are diagonal elements X_{jj}. Also when $\lambda_k > 0$, one can see that with fixed x the diagonal entries of X can be made arbitrarily large to satisfy all constraints. As well when $\lambda_0 > 0$, an arbitrary large diagonal entry of X will push the objective to $-\infty$. Therefore, in general, X_{jj} should be bounded to form a sensible relaxation. In the paper [1] they suppose that $x_j \in [0, 1]$ and use $X_{jj} \leq x_j$ to establish boundedness.

Remark 3.36. In general, if the feasible region is bounded, then there can be introduced box constraints $l_j \leq x_j \leq u_j$ and X_{jj} can be bounded for example by multiplying these two inequalities

$$\left.\begin{array}{r} x_j - l_j \geq 0 \\ u_j - x_j \geq 0 \end{array}\right\} \ x_j u_j + x_j l_j - l_j u_j \geq X_{jj}, \tag{3.37}$$

where $x_j x_j$ was replaced by X_{jj}. This is valid since equality $X = xx^T$ holds for every solution of the original problem.

Following proposition from [11] gives equivalent formulation of $X \succeq xx^T$ constraint, only in terms of x and diagonal entries of X.

Proposition 3.38 ([11]). *Given a vector x and scalars $X_{11}, \dots, X_{nn}$, there exists a symmetric-matrix completion $X \in S^n$ of $X_{11}, \dots, X_{nn}$ satisfying $X \succeq xx^T$ if and only if $X_{jj} \geq x_j^2$ for all $j = 1, \dots, n$.*

Thus, in light of this proposition, the problem with additional bounding constraints $X_{jj} \leq x_j$, the problem (3.35) is equivalent to

$$\begin{array}{ll} \text{minimize} & -\lambda_0 Tr(X) + x^T(P_0 + \lambda_0 I)x + q_0^T x + r_0 \\ \text{subject to} & -\lambda_k Tr(X) + x^T(P_k + \lambda_k I)x + q_k^T x + r_k \leq 0, \\ & (k = 1, \dots, m) \\ & x_j^2 \leq X_{jj} \leq x_j \quad (j = 1, \dots, n) \end{array} \tag{3.39}$$

Compared to SDP relaxation (3.13), which has $O(n^2)$ variables, problem (3.39) has only $O(n)$ and hence is much faster to solve. On the other hand bound should be generally weaker than the SDP bound.

This approach is further generalized and explored in [1].

3.4 LP relaxation of QCQP

Semidefinite program relaxations can provide tight bounds, but they can also be expensive to solve by classical interior point methods (because of the $O(n^2)$ variables and semidefinite constraint).

Many researchers have studied different types of relaxations, for example, ones based on LP or SOCP. As we have seen in the first chapter, SDP is the broadest of the introduced classes, therefore the SDP relaxation will be generally stronger than LP or SOCP relaxation. However the advantage of LP may be the speed.

Simple LP relaxation may be obtained from SDP relaxation (3.13) replacing the semidefinite constraint $X \succeq xx^T$ with symmetry constraint $X = X^T$.

$$\begin{array}{ll} \text{minimize} & P_0 \bullet X + q_0^T x + r_0 \\ \text{subject to} & P_k \bullet X + q_k^T x + r_k \leq 0, \ (k = 1, \ldots, m) \\ & X = X^T. \end{array} \tag{3.40}$$

Suppose that the feasible set in (3.40) is bounded and we can formulate box constraints for each variable - either explicitly included (i.e. $P_k = 0$ for some k) or implied from the quadratic constraints (for example in a combinatorial problem where the constraint $x_i \in \{0,1\}^n$ is included as $x_i(x_i - 1) = 0$ we may also include an additional $0 \leq x_i \leq 1$ box constraint). We will separate all such box constraints in the next formulation.

$$\begin{array}{ll} \text{minimize} & P_0 \bullet X + q_0^T x + r_0 \\ \text{subject to} & P_k \bullet X + q_k^T x + r_k \leq 0, \ (k = 1, \ldots, m) \\ & l_i \leq x_i \leq u_i, \ (i = 1, \ldots, n) \\ & X = X^T. \end{array} \tag{3.41}$$

This is also referred to as lift and project LP relaxation.

3.4.1 Reformulation Linearization Technique (RLT)

The optimal value of (3.41) is usually weak lower bound as no constraint links the values of x and X variables. The main approach to provide these links and strengthen the relaxation is the Reformulation Linearization Technique (RLT) relaxation [15, 16, 17]. It adds linear inequalities to (3.41). These inequalities are derived from the variable bounds and constraints of the original problem as follows: multiply together two original constraints or bounds and relax each product term $x_i x_j$ with the variable X_{ij}. Note that this will be a valid inequality, since the original constraint $X = xx^T$ implies $x_i x_j = X_{ij}$. For instance, let x_i, x_j , $i, j \in \{1, 2, \dots, n\}$ be two variables from (3.41). By taking into account only the four original bounds $x_i - l_i \geq 0$, $x_i - u_i \leq 0$, $x_j - l_j \geq 0$, $x_j - u_j \leq 0$, we get the RLT inequalities

$$\begin{array}{rcl} X_{ij} - u_i x_j - u_j x_i & \geq & -u_i u_j, \\ X_{ij} - u_i x_j - l_j x_i & \leq & -u_i l_j, \\ X_{ij} - l_i x_j - u_j x_i & \leq & -l_i u_j, \\ X_{ij} - l_i x_j - l_j x_i & \geq & -l_i l_j. \end{array} \tag{3.42}$$

Note that these constraints also hold when $i = j$, in which case the upper bounds are identical. Denote $l = (l_1, \dots, l_n)^T$ and $u = (u_1, \dots u_n)^T$. Using the vector and matrix inequalities (meaning that inequality holds in each coordinate) the resulting RLT relaxation can be written as

$$\begin{array}{ll} \text{minimize} & P_0 \bullet X + q_0^T x + r_0 \\ \text{subject to} & P_k \bullet X + q_k^T x + r_k \leq 0, \ (k = 1, \dots, m) \\ & X - lx^T - xl^T \quad \geq \ -ll^T, \\ & X - ux^T - xu^T \ \ \geq \ -uu^T, \\ & X - lx^T - xu^T \ \ \leq \ -lu^T, \\ & l \leq x \leq u, \ \ X = X^T, \end{array} \tag{3.43}$$

Both of the upper bounds for X_{ij} collide together and are expressed in the single vector constraint due to symmetry. In fact, it is known that the original box constraints $l \leq x \leq u$ are redundant and could be removed. If the QCQP contains linear constraints other than simple box constraints (i.e. $P_k = 0$ for some k) then additional constraints on X can be imposed.

3.4.2 Positive semidefinite cuts

In the above LP relaxations we have relaxed nonconvex $X = xx^T$ constraint in (3.6) with simple symmetry and later added RLT constraints to link the variables x and X. In fact we can go further in strengthening the LP relaxation.

Let us begin with SDP relaxation in the form (3.12), i.e.

$$\begin{array}{ll} \text{minimize} & M_0 \bullet Y \\ \text{subject to} & M_k \bullet Y \leq 0, \ (k = 1, \dots, m) \\ & Y \succeq 0, \end{array} \tag{3.44}$$

where

$$M_k = \begin{pmatrix} \alpha_k & \frac{1}{2} q_k^T \\ \frac{1}{2} q_k & P_k \end{pmatrix}, \qquad Y = \begin{pmatrix} 1 & x^T \\ x & X \end{pmatrix}. \tag{3.45}$$

From the definition of positive semidefinite matrix it holds that

$$Y \succeq 0 \ \Leftrightarrow \ v^T Y v \geq 0, \ \forall v \in \mathbb{R}^{n+1}. \tag{3.46}$$

Notice that inequalities $v^T Y v \geq 0$ are linear in Y therefore linear in both x and X, we will refer to the inequalities in this form as PSD cuts. In order to obtain a linear program, the semidefinite constraint cannot be replaced by the infinite number of constraints $v^T Y v \geq 0$. However, it is possible to relax it with the finite number of these linear inequalities.

$$\begin{array}{ll} \text{minimize} & M_0 \bullet Y \\ \text{subject to} & M_k \bullet Y \leq 0, \ (k = 1, \ldots, m) \\ & v^T Y v \geq 0, \ v \in V, \end{array} \tag{3.47}$$

for some finite set of vectors V.

The usual approach is declaring such PSD cuts with regard to a feasible solution $\bar{Y}$ of the relaxation (3.47) (for example $\bar{Y}$ may be the optimal solution of (3.47)). If $\bar{Y}$ is not positive semidefinite, then such vectors v can be added to V, that will cut off $\bar{Y}$ from the feasible set by adding the inequalities $v^T Y v \geq 0$ - that is why it is called PSD cut.

Hopefully, we would like to restrict the feasible set this way to tighten the gap between the optimal value of the LP relaxation (3.47) and the optimal value of the SDP relaxation. For example by cutting off the optimal solutions of the actual LP relaxation.

The vectors v may be chosen as eigenvectors corresponding to negative eigenvalues of an arbitrary matrix $\bar{Y}$. In [16] authors note two weaknesses of such approach. Firstly, only one cut is obtained from each eigenvector, while computing the spectral decomposition requires nontrivial investment of time. Secondly, such cuts are usually very dense, i.e. almost all entries of $\bar{v}\bar{v}^T$ are nonzero, adding such inequalities might considerably slow down the computation.

They propose an efficient algorithm to generate sparse cuts from given matrix $\bar{Y}$ and initial vector $\bar{v}$ with $\bar{v}^T \bar{Y} \bar{v} \leq 0$.

TODO: describe sparsify procedure!

TODO: it might be interesting to find out what is the good choice of such $\bar{Y}$ and $\bar{v}$. (1 - based on problem data, i.e. matrices from quadratic inequalities, 2 - iteratively, always take $\bar{Y}$ as optimal solution of previous relaxation and strengthen by cutting it off, until we reach precision of SDP bound)

3.5 Strengthening the SDP relaxation

In the previous section we talked about loosening the SDP relaxation and providing faster but weaker LP and SOCP relaxations. Even there we have talked about strengthening the resulting relaxations by adding additional valid inequalities (relaxation of semidefinite constraint, RLT or PSD cuts). Here we would like to offer an example of a redundant inequality tightening the relaxation.

TODO!!!

3.5.1 SDP + RLT relaxation

It is not surprising that adding RLT valid inequalities into LP relaxation will significantly strengthen the relaxation. On the other hand, it is quite surprising that RLT inequalities may also help when added to SDP relaxation. In fact Anstreicher in [15] showed that following RLT + SDP relaxation is stronger than both SDP or RLT relaxations.

$$\begin{array}{ll} \text{minimize} & P_0 \bullet X + q_0^T x + r_0 \\ \text{subject to} & P_k \bullet X + q_k^T x + r_k \leq 0, \ (k = 1, \dots, m) \\ & X - lx^T - xl^T \quad \geq \quad -ll^T, \\ & X - ux^T - xu^T \quad \geq \quad -uu^T, \\ & X - lx^T - xu^T \quad \leq \quad -lu^T, \\ & l \leq x \leq u, \quad X \succeq xx^T. \end{array} \tag{3.48}$$

However, in terms of computational cost, in his experiments each RLT or SDP bound for the tested problems required approximately 1 second of computation, but each SDP+RLT bound required over 200 seconds of computation. It is well known that "mixed" SDP/LP problems involving large numbers of inequality constraints are computationally challenging, and reducing the work to solve such problems is an area of ongoing algorithmic research. In [18] they propose enhancing the RLT relaxation with PSD cuts as linear relaxations of the semidefinite constraint. Also adding a SOCP relaxation of semidefinite constraint into lift and project LP relaxation proposed in [4] offers a way to avoid the cost of combining SDP and LP constraints.

3.5.2 Best D.C. decompositions

TODO: complete this subsection!

Another interesting approach for strengthening the SDP relaxation was proposed in [19]. They restrict to the problems with convex quadratic constraints and nonconvex objective, i.e. the matrices $P_1, \dots, P_k$ are assumed to be positive semidefinite.

$$P_k \succeq 0 \ \text{ for } \ k = 1 \dots m. \tag{3.49}$$

Let v_i for $i = 1, \dots, p$ be nonzero vectors in $\mathbb{R}^n$ and suppose that there exists $\lambda \in \mathbb{R}^p_+$ and $\mu \in \mathbb{R}^m_+$ such that

$$P_0 + \sum_{i=1}^{p} \lambda_i v_i v_i^T - \sum_{i=1}^{m} \mu_i P_i \succeq 0, \tag{3.50}$$

Then they decompose the nonconvex objective $f_0(x) = x^T P_0 x + q_0^T x + r_0$ as D.C. (difference of convex functions)

$$x^T \left(P_0 + \sum_{i=1}^{p} \lambda_i v_i v_i^T + \sum_{i=1}^{m} \mu_i P_i \right) x - \sum_{i=1}^{p} \lambda_i (v_i^T x)^2 - \sum_{i=1}^{m} \mu_i x^T P_i x + q_0^T x + r_0. \tag{3.51}$$

After that they convexify the objective by underestimating the nonconvex terms $-\sum_{i=1}^{p} \lambda_i (v_i^T x)^2$ and $-\sum_{i=1}^{m} \mu_i x^T P_i x$ with a (convex) piecewise linear function, and show that optimal choice of parameters λ, μ can be reduced to semidefinite program and show that the resulting SDP has dual equivalent to standard SDP (3.13) strengthened with additional valid inequalities.

3.6 Copositive and completely positive programming representation

Consider a special case of QCQP 1.2 a class of linearly constrained quadratic optimization problems in continuous nonnegative and binary variables

$$\begin{array}{ll} \text{minimize} & x^T Q x + 2c^T x, \\ \text{subject to} & Ax = b, \\ & x_j(1 - x_j) = 0 \ (j \in B), \\ & x \in \mathbb{R}^n_+. \end{array} \tag{3.52}$$

In 2009, completely positive programming (CPP) relaxation for this class of QOPs was proposed by Burer [21]

$$\begin{array}{ll} \text{minimize} & Q \bullet X + 2c^T x, \\ \text{subject to} & Ax = b, \\ & a_i X a_i = b_i^2 \ (i = 1, \dots, n), \\ & x_j = X_{jj} \ (j \in B), \\ & \begin{pmatrix} 1 & x^T \\ x & X \end{pmatrix} \in \mathbb{P}^{n+1}, \end{array} \tag{3.53}$$

where $X \in \mathbb{S}^n$, $x \in \mathbb{R}^n$ are variables, $\mathbb{P}^{n+1}$ is a completely positive cone (see Definition 2.31) and Q, A, c, b, B are from original problem.

As shown in [21] the optimal value of this relaxation coincides with optimal value of (3.52).

The class was extended to a more general class of QOPs by Eichfelder and Povh [22] and by Arima, Kim and Kojima [20]. Theoretically strong results were presented in their papers [20, 21, 22] showing that the exact optimal values of QOPs in their classes coincide with the optimal values of their CPP relaxation problems.

However these CPP relaxations are convex, problem (3.52) involves various NP-hard combinatorial problems (such as Max cut), therefore solving these CPP programs is also NP-hard.

3.6.1 The Lagrangian doubly nonnegative relaxation

In this section we will describe relaxation by Kim, Kojima and Toh [23] of the more general problem with complementarity constraints

$$\begin{array}{ll} \text{minimize} & u^T Q u + 2c^T u, \\ \text{subject to} & Au = b, \\ & u_i u_j = 0, \ ((i,j) \in \mathcal{E}), \\ & u \in \mathbb{R}^n_+, \end{array} \tag{3.54}$$

where $u \in \mathbb{R}^n_+$ is a variable, matrices $Q \in \mathbb{S}^n$, $A \in \mathbb{R}^{m\times n}$, vectors $c \in \mathbb{R}^n$, $b \in \mathbb{R}^m$ and set $\mathcal{E} \subset \{(i,j) \mid 1 \le i < j \le n\}$ are given problem data. Note that binary constraint $u_i(1-u_i) = 0$ can be converted to complementary $u_i v_i = 0$ by introducing slack variable $v_i = (1 - u_i) \ge 0$. In [23] authors assume that linear constraint set $\{u \in \mathbb{R}^n_+ \mid Au = b\}$ is bounded.

The problem (3.54) is first reformulated as

$$\begin{array}{ll} \text{minimize} & Q_0 \bullet xx^T, \\ \text{subject to} & H_0 \bullet xx^T = 1, \\ & Q_{01} \bullet xx^T = 0, \\ & Q_{ij} \bullet xx^T = 0, \ ((i,j) \in \mathcal{E}), \\ & x \in \mathbb{R}^{n+1}_+, \end{array} \tag{3.55}$$

where

$$x = (x_1, u) \in \mathbb{R}^{n+1}_+ \tag{3.56}$$

$$Q_0 = \begin{pmatrix} 0 & c^T \\ c & Q \end{pmatrix} \in \mathbb{S}^{n+1}, \tag{3.57}$$

$$H_0 = \begin{pmatrix} 1 & 0^T \\ 0 & O \end{pmatrix} \in \mathbb{S}^{n+1}, \tag{3.58}$$

$$Q_{01} = \begin{pmatrix} b^T b & b^T A \\ a^T b & A^T A \end{pmatrix} \in \mathbb{S}^{n+1}, \tag{3.59}$$

$$Q_{ij} = \begin{pmatrix} 0 & 0^T \\ 0 & C_{ij} + Cji \end{pmatrix} \in \mathbb{S}^{n+1}, \tag{3.60}$$

$$C_{ij} = \begin{array}{l} \text{the } n \times n \text{ matrix with } (i,j)\text{-th element } 1/2 \\ \text{and } 0 \text{ elsewhere.} \end{array} \tag{3.61}$$

Since the matrices Q_k for $k = 1, \ldots, m$ are nonegative, $Q_k \bullet xx^T \ge 0$ holds for any $x \in \mathbb{R}^n_+$. Hence, the set of inequalities $Q_k \bullet xx^T = 0$ can be combined into a single equality $H_1 \bullet xx^T = 0$, where $H_1 = \sum_{k=1}^m Q_k$. Consequently they obtain simplified problem

$$\begin{array}{ll} \text{minimize} & Q_0 \bullet xx^T, \\ \text{subject to} & H_0 \bullet xx^T = 1, \\ & H_1 \bullet xx^T = 0, \\ & x \in \mathbb{R}^n_+, \end{array} \tag{3.62}$$

which is equivalent to (3.54). Here we can equivalently replace xx^T by $X \in \Gamma$ (recall that $\Gamma = \{xx^T \mid x \in \mathbb{R}^n_+\}$). This problem is in [23] relaxed as

$$\begin{array}{ll} \inf & Q_0 \bullet X, \\ \text{subject to} & H_0 \bullet X = 1, \\ & H_1 \bullet X = 0, \\ & X \in \mathcal{K}, \end{array} \tag{3.63}$$

where $\Gamma \subseteq \mathcal{K}$. The cone $\mathcal{K}$ can be chosen as any of the Γ, $\mathbb{P}^n$, $\mathbb{S}^n \cap \mathbb{N}^n$, $\mathbb{S}^n$, $\mathbb{S}^n + \mathbb{N}^n$, $\mathbb{C}^n$. However, it is usually chosen as $\mathbb{S}^n \cap \mathbb{N}^n$ resulting in doubly nonnegative (DNN) relaxation.

The idea behind this particular reformulation of 3.54 is that its Lagrangian relaxation

$$L^p(\lambda, \mathcal{K}) := \inf\{Q_0 \bullet X + \lambda H_1 \bullet X \mid H_0 \bullet X = 1,\ X \in \mathcal{K}\} \tag{3.64}$$
$$L^d(\lambda, \mathcal{K}) := \sup\{y \mid Q_0 + \lambda H_1 - y_0 H_0 \in \mathcal{K}^*\} \tag{3.65}$$

has dual with only one variable. As they show in [23] when $\mathcal{K}$ is a closed cone such that $\Gamma \subseteq \mathcal{K} \subseteq \mathbb{S}^n \cap \mathbb{N}^n$, then for a sufficiently large λ, the optimal value of this relaxation coincides with the optimal value of (3.54).

That is also the reason why we can write maximize and minimize instead of inf and sup when $\mathcal{K} = \mathbb{S}^n \cap \mathbb{N}^n$ is chosen, forming a Lagrangian doubly nonnegative (DNN) relaxation of (3.54).

Solving a DNN relaxation using a standard SDP solver is computationally costly compared to SDP relaxation due to the high number of linear (nonnegative) constraints. In [23] authors Kim, Kojima and Toh proposed a fast algorithm for solving Lagrangian DNN relaxation (3.64 - 3.65) based on combination of bisection method and first order methods. Recently (February 2016), Arima, Kim, Kojima and Toh [24] proposed an improvement of the algorithm which achieves better robustness and acceleration. Their approach has been shown to provide tight bounds for combinatorial problems as binary QPs, multiple knapsack problems, maximal stable set problems and the quadratic assignment problems.

3.7 Lasserre hierarchy

Earlier in this chapter we have mentioned approaches for strengthening the LP and SDP relaxations (as PSD cuts, RLT or adding nonnegative cone constraint). Instead of following the heuristic approach of finding a valid inequalities that may be helpful for strengthening an LP or SDP, there is a more systematic (and potentially more powerful) approach lying in the use of LP or SDP hierarchies. In particular there are procedures by Balas, Ceria, Cornuéjols [27]; Lovász, Schrijver [28] (with LP-strengthening LS and an SDP-strengthening LS+); Sherali, Adams [29] or Lasserre [30, 31].

In this section we will describe the Lasserre hierarchy of successive SDP relaxation, which has been shown to be superior and to produce tightest relaxations in the comparison by Laurent [32].

Consider the following binary polynomial optimization problem

$$\begin{array}{ll} \text{minimize} & p_0(x), \\ \text{subject to} & p_k(x) \le 0, \ (k = 1, \dots, m) \\ & x \in \{0,1\}^n. \end{array} \tag{3.66}$$

Let $K = \{x \in \mathbb{R}^n \ p_k(x) \le 0, \ k = 1, \dots, m\}$. Strenghtening a relaxation of K, is done by adding additional variables and constraints in order to make this relaxation tighter. The final (and unreachable) goal is to have a convex relaxation of the feasible region exactly $conv(K \cap 0, 1^n)$ (the best possible convex relaxation). The idea is enforcing a local constraints on greater neighbourhoods in each round.

Let us first introduce the notation. The r-th degree polynomial $p(x) : \mathbb{R}^n \to \mathbb{R}$ in basis

$$1, x_1, \ \dots, \ x_n, \ x_1^2, \ x_1x_2, \ \dots, \ x_n^2, \ \dots, \ x_1^r, \ \dots, \ x_n^r, \tag{3.67}$$

is written as

$$p(x) = \sum_\alpha x^\alpha p_\alpha, \qquad \text{where} \ \ x^\alpha = x_1^{\alpha_1} \cdots x_n^{\alpha_n}.$$

Denote $p = \{p_a\}$ the vector of coefficients of the $p(x)$ in the basis (3.67). Hence the vectors of coefficients of the polynomials $p_k(x)$ in (3.66) are p_k for $k = 1, \dots, m$.

Denote $V = \{1, \dots, n\}$, $\mathcal{P}(V)$ the collection of all subsets of V and $\mathcal{P}_t(V)$ for $1 \le t \le n$ the collection of all subsets of cardinality $\le t$. Next we will define important notions of moment matrix and localizing matrix.

For $x, y \in \mathbb{R}^{\mathcal{P}(V)}$, the $x * y$ denotes a vector of $\mathbb{R}^{\mathcal{P}(V)}$ with entries

$$(x * y)_I := \sum_{K \subseteq V} x_K y_{I \cup K}. \tag{3.68}$$

Definition 3.69 (Moment and Localizing matrix)**.** Given a vector $y \in \mathbb{R}^{\mathcal{P}(V)}$ and an integer t, $1 \le t \le n$ the matrices

$$\begin{aligned} M_t(y) &:= (y_{I \cup J})_{|I|,|J| \le t} && (3.70) \\ M_t^k(y) &:= ((p_k * y)_{I \cup J})_{|I|,|J| \le t}, \ \forall \ k = 1, \dots, m, && (3.71) \end{aligned}$$

are known as moment matrix and localizing matrices respectively.

The Lasserre relaxations are based on following observation

Lemma 3.72. *Given $x \in K \cap \{0,1\}^n$, the vector $y \in \mathbb{R}^{\mathcal{P}(V)}$ with entries $y_I := \prod_{i \in I} x_i$, satisfies*

$$M_t(y) \succeq 0, \quad M_t^k(y) \succeq 0, \ \forall \ k = 1, \dots, m.$$

Proof. Indeed, for any $I, J \subseteq V$ with $|I| \le 2t$, it holds that

$$\begin{aligned} [M_t(y)]_{I,J} &= [yy^T]_{I,J} \\ \left[M_t^k(y)\right]_{I,J} &= \left[(p_k * y)(p_k * y)^T\right]_{I,J} \end{aligned}$$

since $y_{I \cup J} = y_I y_J$ for all $I, J \subseteq V$. □

Definition 3.73 (Lasserre Hierarchy)**.** Let $K = \{x \in \mathbb{R}^n \mid p_k(x) \leq 0,\ k = 1, \ldots, m\}$. We define t-th round of Lasserre relaxation $L_t(K)$ as the set of vectors $y \in \mathbb{R}^{\mathcal{P}(V)}$ that satisfy

$$\begin{array}{rcl} y_\emptyset & = & 1, \\ M_t(y) & \succeq & 0, \\ M_t^k(y) & \succeq & 0,\ \forall\ k = 1, \ldots, m. \end{array} \tag{3.74}$$

Especially, when all the constraints are linear, i.e. $K = \{x \in \mathbb{R}^n \mid Ax \geq b\}$, we have

$$\begin{array}{rcl} y_\emptyset & = & 1, \\ M_t(y) & = & (y_{I \cup J})_{|I|,|J| \leq t} \succeq 0, \\ M_t^k(y) & = & \left(\sum_{i=1}^n A_{ki} y_{I \cup j \cup \{i\}} - b_k y_{I \cup J}\right) \succeq 0, \quad \forall k = 1, \ldots m. \end{array} \tag{3.75}$$

The relaxed problem then takes form

$$\begin{array}{ll} \text{minimize} & \sum_\alpha (p_0)_\alpha y_\alpha \\ \text{subject to} & y \in L_t(K). \end{array} \tag{3.76}$$

Let us denote p_t^* the optimal value of t-th round Lasserre relaxation. In [30] Lasserre shows that the sequence p_t^* converges to optimal value of the original problem (3.66). What is more, the convergence is finite.

In fact, Lasserre states that every program (3.66) is equivalent to SDP with $2^m - 1$ (because any $L_t(K)$ can be reduced to $2^m - 1$ variables). And the projection of the feasible set $L_m(K)$ onto space spanned by the n variables $y_{\{1\}}, \ldots, y_{\{n\}}$ is the convex hull of K.

So far the Lasserre hierarchy does not seem like a practical tool for solving combinatorial problems because of the rapidly growing complexity, but has a potential to offer valuable insights. In the survey [33] they discuss an application in approximation algorithms. For example for the maximum cut problem they prove equivalence of the 3rd round of Lasserre with the famous SDP relaxation by Goemans and Williamson [37] (see 4.25).

On the other hand, the hierarchy of Lasserre might be a good measure of hardness of problem instances. In [**??**] authors discuss hard instances of binary problems and state a conjecture that at least $\lceil \frac{n}{2} \rceil$ rounds are needed to obtain exact relaxation for max cut problem.

Chapter 4

The maximum cut problem

In this chapter we are going to explore one of the classical problems of combinatorial optimization - the maximum cut problem. First we will state the problem and show that it can be formulated as QCQP. Then we will describe in detail the famous analysis by Goemans and Williamson [37] showing its SDP relaxation gives 0.878 approximation bound.

In the end of this section we will describe how each of the relaxations mentioned in the Chapter 3 can be applied to max cut problem. And provide a computational comparison of these approaches.

We have chosen this particular problem for two reasons. First reason is kind of historical, because the tight bound for the semidefinite relaxation by Goemans and Williamson [37] seems to be the initial motivation for the extensive research of semidefinite relaxations, algorithms for semidefinite programming and convex relaxations of other combinatorial problems. The second reason is the simplicity of the problem formulation and ease of adapting the general relaxation methods for QCQP.

Since the semidefinite relaxation of max cut is somehow considered as benchmark for evaluating the performance of other relaxation methods, most of the approaches mentioned in the Chapter 3 have been already computationally compared with SDP relaxation on the instances of the max cut problem. However, we believe that an cross comparison of all these relaxation might be also an insightful observation.

4.1 Problem formulation

Problem statement (max-cut). Let $G = (V, \mathcal{E})$ be an undirected graph where $V = \{1, \ldots, n\}$ and $\mathcal{E}$ are the sets of vertices and edges, respectively. We assume that a weight w_{ij} is attached to each edge $[i, j] \in \mathcal{E}$. For a partition $(S, \bar{S})$ of V, i.e.

$$(S, \bar{S}), \quad \text{s.t.} \quad S \cup \bar{S} = V \quad \wedge \quad S \cap \bar{S} = \emptyset, \tag{4.1}$$

we define

$$w(S, \bar{S}) = \sum_{[i,j] \in \mathcal{E}, i \in S, j \in \bar{S}} w_{ij}. \tag{4.2}$$

The maximum cut problem (or shortly max-cut) is to find a partition maximizing $w(S, \bar{S})$.

4.1.1 QCQP formulation of the max-cut

Let us assign zero weight to all the non edges $w_{ij} = 0$ for $[i, j] \notin \mathcal{E}$. For each $i \in V$, we put

$$x_i = \begin{cases} 1 & \text{if } i \in S, \\ -1 & \text{if } i \in \bar{S}. \end{cases} \tag{4.3}$$

Because $\frac{1}{2}(1 - x_i x_j) = 1$ if i and j belong to different partitions and 0 otherwise, we see that problem is equivalent to

$$\begin{array}{ll} \text{maximize} & \frac{1}{2}\sum_{i<j} w_{ij}(1 - x_i x_j), \\ \text{subject to} & x \in \{-1, 1\}^n. \end{array} \tag{4.4}$$

This can be easily rewritten in matrix form. Let W be a $n \times n$ symmetric matrix of weights with entries $W_{ij} = w_{ij}$, and let

$$L = diag(We) - W, \tag{4.5}$$

where e is all ones vector and $diag(We)$ is the diagonal matrix with diagonal We. Then, using $x_i^2 = 1$ we have

$$\begin{aligned} w(S, \bar{S}) &= \frac{1}{2}\sum_{i<j} w_{ij}(1 - x_i x_j) && (4.6) \\ &= \frac{1}{4}\sum_{[i,j] \in V^2} w_{ij}(1 - x_i x_j) && (4.7) \\ &= \frac{1}{4} x^T (diag(We) - W) x && (4.8) \\ &= \frac{1}{4} x^T L x. && (4.9) \end{aligned}$$

Now, with $x_i \in \{0, 1\} \Leftrightarrow x_i^2 = x^T e_i e_i^T x = 1$ constraint, we obtain a following problem

$$\begin{array}{ll} \text{maximize} & \frac{1}{4} x^T L x, \\ \text{subject to} & x^T e_i e_i^T x = 1, \ (i = 1, \ldots, n). \end{array} \tag{4.10}$$

Therefore max-cut is indeed an instance of QCQP.

4.2 The Goemans and Williamson analysis

In their famous article [37] authors first relax the problem (4.4) by allowing x_i to be represented by vector variables on the unit sphere $v_i \in S^n$ for $i \in V$. And define objective function by replacing the $x_i x_j$ with scalar products $v_i^T v_j$. In this way, the objective reduces to $\frac{1}{2}\sum_{i<j} w_{ij}(1 - x_i x_j)$ when all the v_i are lying in 1-dimensional space. The resulting relaxation is

$$\begin{array}{ll} \text{maximize} & \frac{1}{2}\sum_{i<j} w_{ij}(1 - v_i^T v_j), \\ \text{subject to} & v_i \in S^n \ (i = 1, \ldots, n). \end{array} \tag{4.11}$$

We will show later that this relaxation can be solved using the semidefinite programming.

They propose simple randomized algorithm, which we will refer to as GW-algorithm

1. Solve (4.11), obtaining an optimal set of vectors v_i.
2. Generate r uniformly distributed on the unit sphere S^n.
3. Set $S = \{i \mid v_i^T r \geq 0\}$.

In other words in step 2, the random hyperplane through the origin is chosen. In step 3 we form a partitions based on the separation of the vectors v_i by the hyperplane. The sets S and $\bar{S}$ are formed by indices of vectors lying "above" and "bellow" the hyperplane, respectively.

Proposition 4.12. *Let ω be the value of the cut produced by the above algorithm and $E[\omega]$ its expected value. Authors show that*

$$E[\omega] = \frac{1}{\pi}\sum_{i<j} w_{ij} \arccos(v_i^T v_j), \tag{4.13}$$

and

$$E[\omega] \geq \alpha\frac{1}{2}\sum_{i<j} w_{ij}(1 - v_i^T v_j) \quad \textit{for} \ \ \alpha = 0.87856. \tag{4.14}$$

Proof. First we will show (4.13). By definition and linearity of expected value in the first step, using symmetry in the second we have

$$\begin{aligned} E[\omega] &= \sum_{i<j} w_{ij}\ Pr\left[sgn(v_i^T r) \neq sgn(v_j^T r)\right] && (4.15) \\ &= \sum_{i<j} w_{ij}\ 2Pr\left[v_i^T r \geq 0\ \wedge\ v_j^T r < 0\right]. && (4.16) \end{aligned}$$

Fix the i, j and let $\theta = \arccos(v_i^T v_j)$ be the angle between v_i and v_j. In the last expression, there is a probability that v_i is above the hyperplane with normal vector r and v_j is bellow. The set $\{r \mid v_i^T r \geq 0\ \wedge\ v_j^T r < 0\}$ corresponds to the intersection of two half-spaces whose dihedral angle is precisely θ its intersection with the sphere is a spherical digon of angle θ and, by symmetry of the sphere, thus has measure equal to $\theta/2\pi$ times the measure of the full sphere. In other words,

$$Pr\left[v_i^T r \geq 0\ \wedge\ v_j^T r < 0\right] = \frac{\theta}{2\pi} = \frac{\arccos(v_i^T v_j)}{2\pi}. \tag{4.17}$$

Plugging this identity into (4.16), the first part follows.

Secondly we will show (4.14). It is enough to prove that

$$\frac{1}{\pi}\arccos(v_i^T v_j) \geq \alpha\frac{1}{2}(1 - v_i^T v_j), \tag{4.18}$$

because multiplying by w_{ij} and summing up these inequalities yelds

$$E[\omega] = \frac{1}{\pi} \sum_{i<j} w_{ij} \arccos(v_i^T v_j) \geq \alpha \frac{1}{2} \sum_{i<j} w_{ij}(1 - v_i^T v_j), \tag{4.19}$$

where the left hand side is equal to $E[\omega]$ from (4.13).

Since $v_i, v_j \in S^n$ are unit vectors, the inner product is bounded by $v_i^T v_j \in [-1, 1]$. Thus, there exists $\theta \in [0, \pi]$ such that $v_i^T v_j = \cos(\theta)$. Rearranging terms and substituting $v_i^T v_j = \cos(\theta)$ we can see that (4.18) reduces to

$$0.87856 = \alpha \leq \min_{\theta \in [0,\pi]} \frac{2}{\pi} \frac{\theta}{1 - \cos(\theta)}. \tag{4.20}$$

Which is just a simple exercise to prove.

□

Remark 4.21. Denote the objective values

ω_r	–	value in the optimal solution of the relaxation (4.11)
ω^*	–	value in the optimal solution of the maxcut problem (4.4)
ω	–	value in the feasible solution of (4.4) obtained by the GW-algorithm

Then the following inequality holds trivially,

$$\omega \leq \omega^* \leq \omega. \tag{4.22}$$

The proposition 4.12 further implies

$$0.878\ \omega_r < E[\omega] \leq \omega^* \leq \omega_r. \tag{4.23}$$

Therefore, the bounds obtained by solving the relaxed problem are quite tight. What is more, the feasible solutions projected by GW-algorithm are also good in expectation. Generally, we need to solve the relaxation (4.11) only once and then repeat steps 2 and 3 of the algorithm couple of times. Keeping only the best solution one will (probably) get the objective at least as good as the expected value.

4.2.1 Semidefinite relaxation of the max-cut

Now the last arising question is how do we solve the relaxed problem (4.11)? Let us remind the formulation of the relaxed problem

$$\begin{array}{ll} \text{maximize} & \frac{1}{2}\sum_{i<j} w_{ij}(1 - v_i^T v_j), \\ \text{subject to} & v_i \in S^n \ (i = 1, \ldots, n), \end{array} \tag{4.24}$$

where $v_i \in S^n$ are the variables, the scalar weights $w_{ij} \in \mathbb{R}_+$ are given and the indices i, j are in $\{1, \ldots, n\}$.

Let $B := (v_1, \ldots, v_n)$ be the $n \times n$ matrix with vectors v_i as columns. It holds that matrix $X := BB^T$ has elements $X_{ij} = v_i^T v_j$ for each $i, j \in (1, \ldots, n)$, and X is positive semidefinite. Since $v_i \in S^n$ are unit vectors, the diagonal entries of X are ones, $X_{ii} = v_i^T v_i = 1$. We can find such X by solving following SDP and extract unit vectors v_i from Choelsky decomposition $X = BB^T$.

$$\begin{array}{ll} \text{maximize} & \frac{1}{4} L \bullet X, \\ \text{subject to} & diag(X) = e \\ & X \succeq 0, \end{array} \tag{4.25}$$

where $X \in \mathbb{S}_+^n$ is a variable, the matrix L is defined in (4.5), $diag(X)$ is the vector of diagonal entries of X, and e is the n-dimensional vector of ones.

Remark 4.26. In fact, we would receive the same semidefinite relaxation by following the general scheme for QCQP described in Section 3.1.

4.3 Relaxations of the max cut

4.3.1 Semidefinite relaxation

W will derive the SDP relaxation for max cut once again, now using the procedure for QCQP described in Section 3.1.

First introduce a new variable $X = xx^T$ in (4.4) and rewrite this constraint as $X \succeq 0$, and rank $X = 1$. The maximum cut problem is then equivalent to

$$\begin{array}{ll} \text{maximize} & \frac{1}{4} L \bullet X, \\ \text{subject to} & e_i e_i^T \bullet X = 1, \ (i = 1, \dots, n), \\ & X \succeq 0, \quad \text{rank } X = 1. \end{array} \tag{4.27}$$

However, each of the constraints $e_i e_i^T \bullet X = 1$ is equivalent to $X_{ii} = 1$, altogether giving $diag(X) = e$, here $diag(X)$ denotes the diagonal of the matrix X. What is more, relaxing the nonconvex rank 1 constraint we obtain the SDP relaxation

$$\begin{array}{ll} \text{maximize} & \frac{1}{4} L \bullet X, \\ \text{subject to} & diag(X) = e \\ & X \succeq 0. \end{array} \tag{4.28}$$

4.3.2 Triangle inequalities

In the above reformulation of max cut (4.27), it is clear that following inequalities hold for X and any triple of vertices $i, j, k \in V$

$$\begin{array}{rcl} X_{ij} + X_{jk} + X_{ik} & \geq & -1, \\ X_{ij} - X_{jk} - X_{ik} & \geq & -1, \\ -X_{ij} - X_{jk} + X_{ik} & \geq & -1, \\ -X_{ij} + X_{jk} - X_{ik} & \geq & -1 \end{array} \tag{4.29}$$

since at least two of i, j, k should be contained in the same partition. These inequalities are called triangle inequalities and play an important role in strengthening of max cut relaxations.

Adding them into SDP relaxation (or into any relaxation with matrix variable $X = xx^T$) usually results in better bounds but on the other hand, large number of linear constraints will also increase the computational time.

4.3.3 SOCP relaxations

Let

$$\lambda_1 \le \dots \le \lambda_l \le 0 < \lambda_{l+1} \le \dots \ge \lambda_n,$$

be the eigenvalues of matrix L and let $q_1, \dots, q_n$ be the corresponding unit eigenvectors. Denote $L^- = -\sum_{j=1}^{l} \lambda_j q_j q_j^T \succeq 0$.

Using the framework by Kim and Kojima [4] which we have described in the Section 3.2.3 we obtain following SOCP relaxation of max cut

$$\begin{array}{ll} \text{maximize} & t, \\ \text{subject to} & \frac{1}{4}x^T L^- x + \sum_{j=l+1}^{n} \lambda_j z_j + t \le 0, \\ & x^T q_j q_j^T x - z_j \le 0, \ (j = l+1, \dots, n), \\ & x^T e_j e_j^T x \le 1, \ (j = 1, \dots, n), \\ & z_j \le \sqrt{n}, \ (j = l+1, \dots, n). \end{array} \tag{4.30}$$

Here, the bound for $z_j = q_j^T X q_j$ comes from the fact that X_{ij} is either $+1$ or -1, and $\|q\|_2 = 1$.

In fact, if G has no loops, i.e. $w_{ii} = 0 \ \forall i \in V$ and all weights are nonnegative, then L is diagonally dominant, since

$$L_{ii} = \sum_{j \in V} w_{ij} = \sum_{i \ne j} |L_{ij}|.$$

Hence, L is positive semidefinite and $L^- = 0$. Therefore the quadratic term in the first constraint vanishes.

An improvement for max cut

Authors Muramatsu and Suzuki [38] propose an improvement of this relaxation (especially for the max cut) based on the problem structure. Their approach considers different choice of the matrices C in the relaxation of semidefinite constraint $C \bullet X - x^T C x \ge 0$.

Denote

$$\begin{aligned} u_{ij} &= e_i + e_j, \\ v_{ij} &= e_i - e_j. \end{aligned}$$

Their choice of matrices C consist of the following

$$e_i e_i^T, \qquad i = 1, \dots, n, \tag{4.31}$$
$$u_{ij} u_{ij}^T, \qquad [i,j] \in \mathcal{E}, \tag{4.32}$$
$$v_{ij} v_{ij}^T, \qquad [i,j] \in \mathcal{E}, \tag{4.33}$$

where $\mathcal{E}$ denotes the set of edges, in terms of entries of matrix L the set $\mathcal{E} = \{[i,j] \mid i,j \in V,\ i \neq j,\ L_{ij} \neq 0\}$. The corresponding quadratic constraints are

$$\begin{aligned} x^T e_i e_i^T x - e_i e_i^T \bullet X &\leq 0,\ i = 1,\dots,n, && (4.34)\\ x^T u_{ij} e_{ij}^T x - u_{ij} u_{ij}^T \bullet X &\leq 0,\ [i,j] \in \mathcal{E}, && (4.35)\\ x^T v_{ij} e_{ij}^T x - u_{ij} v_{ij}^T \bullet X &\leq 0,\ [i,j] \in \mathcal{E}. && (4.36) \end{aligned}$$

As Kim and Kojima in [4], they also reduce the number of variables. In fact, since $X_{ii} = 1$, the constraint (4.34) reduces to $x_i^2 \leq 1$. By introducing the new variables

$$\begin{aligned} s_{ij} &:= u_{ij}^T X u_{ij},\ [i,j] \in \mathcal{E}, && (4.37)\\ z_{ij} &:= v_{ij}^T X v_{ij},\ [i,j] \in \mathcal{E}, && (4.38) \end{aligned}$$

they obtain a quadratic inequalities from (4.35) and (4.36)

$$\begin{aligned} (x_i + x_j)^2 \leq s_{ij},\ [i,j] \in \mathcal{E}, && (4.39)\\ (x_i - x_j)^2 \leq z_{ij},\ [i,j] \in \mathcal{E}. && (4.40) \end{aligned}$$

For those variables they have following bound

$$s_{ij} + z_{ij} = X \bullet (u_{ij} u_{ij}^T - v_{ij} v_{ij}^T) = 2(X_{ii} + X_{jj}) = 4. \tag{4.41}$$

Furthermore, they prove following proposition

Proposition 4.42.

$$L = -\sum_{[i,j]\in\mathcal{E}} L_{ij} v_{ij} v_{ij}^T. \tag{4.43}$$

Proof. Let

$$\delta_{ij} = e_i^T e_j = \begin{cases} 1, & \text{if } i = j,\\ 0, & \text{otherwise.} \end{cases}$$

The (k,l) th componen of the right-hand side of (4.43) is

$$\begin{aligned} e_k^T \left(-\sum_{[i,j]\in\mathcal{E}} L_{ij} v_{ij} v_{ij}^T \right) e_l &= -\sum_{[i,j]\in\mathcal{E}} L_{ij} e_k^T (e_i - e_j)(e_i - e_j)^T e_l\\ &= -\sum_{[i,j]\in\mathcal{E}} L_{ij} (\delta_{ki}\delta_{il} + \delta_{kj}\delta_{jl} - \delta_{ki}\delta_{jl} - \delta_{kj}\delta_{il})\\ &= \begin{cases} -\sum_{[k,j]\in\mathcal{E}} L_{kj} & \text{if } k = l,\\ L_{kl} & \text{if } [k,l] \in \mathcal{E},\\ 0 & \text{otherwise.} \end{cases}\\ &= L_{kl} \end{aligned}$$

□

Using the proposition the objective is rewritten as

$$\frac{1}{4} L \bullet X = -\frac{1}{4} \sum_{[i,j]\in\mathcal{E}} L_{ij} v_{ij} v_{ij}^T \bullet X = -\frac{1}{4} \sum_{[i,j]\in\mathcal{E}} L_{ij} v_{ij}^T X v_{ij} = -\sum_{[i,j]\in\mathcal{E}} L_{ij} z_{ij}. \tag{4.44}$$

Now with X removed they obtain the following relaxation

$$\begin{array}{ll} \text{maximize} & -\frac{1}{4}\sum_{[i,j]\in\mathcal{E}} L_{ij} z_{ij}, \\ \text{subject to} & x_i^2 \leq 1,\ (i=1,\ldots,n), \\ & (x_i+x_j)^2 \leq s_{ij},\ [i,j]\in\mathcal{E}, \\ & (x_i-x_j)^2 \leq y_{ij},\ [i,j]\in\mathcal{E}, \\ & s_{ij}+z_{ij}=4,\ [i,j]\in\mathcal{E}. \end{array} \tag{4.45}$$

They also propose adding a certain form of triangle inequalities into relaxation (4.45).

Proposition 4.46. *If $[i,j],\ [j,k],\ [k,l] \in \mathcal{E}$, then following inequalities are valid*

$$\begin{array}{rcl} z_{ij}+z_{jk}+z_{ki} & \leq & 8, \\ z_{ij}+s_{jk}+s_{ki} & \leq & 8, \\ s_{ij}+s_{jk}+z_{ki} & \leq & 8, \\ s_{ij}+z_{jk}+s_{ki} & \leq & 8. \end{array}$$

Proof. Since $X_{ii}=1$, we have

$$\begin{array}{rcl} z_{ij}+z_{jk}+z_{ki} & = & v_{ij}^T X v_{ij} + v_{jk}^T X v_{jk} + v_{ki}^T X v_{ki} \\ & = & 2(X_{ii}+X_{jj}+X_{kk}) - 2(X_{ij}+X_{jk}+X_{ki}) \\ & = & 6-2(X_{ij}+X_{jk}+X_{ki}). \end{array}$$

From triangle inequality $X_{ij}+X_{jk}+X_{ki} \geq -1$ it follows that $z_{ij}+z_{jk}+z_{ki} \leq 8$. The other three inequalities can be proved similarly. □

Putting all together (triangle inequalities and (4.45) we have third SOCP relaxation for max cut

$$\begin{array}{ll} \text{maximize} & -\frac{1}{4}\sum_{[i,j]\in\mathcal{E}} L_{ij} z_{ij}, \\ \text{subject to} & x_i^2 \leq 1,\ (i=1,\ldots,n), \\ & \left.\begin{array}{l} (x_i+x_j)^2 \leq s_{ij}, \\ (x_i-x_j)^2 \leq y_{ij}, \\ s_{ij}+z_{ij}=4, \end{array}\right\} [i,j]\in\mathcal{E}, \\ & \left.\begin{array}{rcl} z_{ij}+z_{jk}+z_{ki} & \leq & 8, \\ z_{ij}+s_{jk}+s_{ki} & \leq & 8, \\ s_{ij}+s_{jk}+z_{ki} & \leq & 8, \\ s_{ij}+z_{jk}+s_{ki} & \leq & 8. \end{array}\right\} [i,j],\ [j,k],\ [k,l]\in\mathcal{E}. \end{array} \tag{4.47}$$

As the number of the variables in these relaxations (4.45), (4.47) depend highly on the number of edges in $\mathcal{E}$, it is expected that these relaxations will do better on the sparse graphs. Authors of [38] also propose fixing the nodes with high degree first as a heuristic for branching in the branch and bound procedure in order to obtain a speed up by decreasing the number of variables.

4.3.4 Mixed SOCP-SDP relaxation

4.3.5 LP relaxations

Standard LP relaxation

The standard relaxation is easily obtained form the SDP relaxation (4.49) omitting the semidefinite constraint

$$\begin{array}{ll} \text{maximize} & \frac{1}{4}L \bullet X, \\ \text{subject to} & diag(X) = e, \\ & X = X^T. \end{array} \tag{4.48}$$

This relaxation is usually strengthened by adding the triangle inequalities (4.29) resulting in

$$\begin{array}{lrcl} \text{maximize} & \frac{1}{4}L \bullet X, & & \\ \text{subject to} & diag(X) = e, & & \\ & X = X^T, & & \\ & X_{ij} + X_{jk} + X_{ik} & \geq & -1, \\ & X_{ij} - X_{jk} - X_{ik} & \geq & -1, \\ & -X_{ij} - X_{jk} - X_{ik} & \geq & -1, \\ & -X_{ij} + X_{jk} - X_{ik} & \geq & -1. \end{array} \tag{4.49}$$

LP with RLT relaxation

Are ther some constraints we can use for RLT? (triangle?) If yes, add them also to SDP.

LP with PSD cuts relaxation

4.3.6 DNN relaxations

For these relaxations we will need to reformulate max cut as a problem with binary 0-1 variables.

DNN relaxation

Lagrangian DNN relaxation

Chapter 5

Computational comparison of the relaxation methods

A great comparison of various LP and SDP branch and bound algorithms for the minimum graph bisection problem was done in [36].

Next we will introduce the branch and bound framework, which provides a way to solve (not only) combinatorial problems to optimality using bounds obtained by solving relaxed problems.

5.1 Branch and bound

We will first describe a general idea of the branch and bound algorithms. uppose we are dealing with minimization problem.

The basic idea of this method is to divide and conquer. There are 3 basics steps we will do repeatedly. Branch, compute bounds and prune.

- Branch. First divide by branching the original problem into the smaller subproblems i.e. by partitioning the feasible set. In combinatorial (discrete) problems this can be done by fixing a variable. For example if $x_l \in \{0, 1\}$, then by fixing $x_l = 1$ and $x_l = 0$ respectively, the problem is divided into two subproblems with one less free variable.

  Although, the branch and bound algorithms are mostly developed and used for solving combinatorial problems, it is also possible to adapt then for general QCQP. Whenever the feasible set is compact, the branching can be done by subdividing the feasible region into the Cartesian product of triangles and rectangles (see [34, 35]).

- Compute bounds. The lower bounds should be established for each branch. Usually this is done by solving the relaxed problem for each subproblem. An upper bound is computed on the optimal value, this can be done by finding a feasible solution for chosen subproblem. It is often extracted from the solution of the relaxed problem using a projection or rounding procedure.

- Prune. Having both lower and upper bounds, conquer by pruning all the branches with lower bound greater then global upper bound. The optimal solution is surely not in these branches.

- Repeat. This process is repeated by branching the remaining subproblems, computing the new (better) lower bounds for these smaller problems, improving the upper bound and further pruning the problem tree.

One can use more or less clever techniques to decide when, how and which subproblems to branch or how and when to compute bounds. Using stronger relaxations for bounding steps results in increased computation time, but hopefully less iterations are needed since better bounds allow to cut branches earlier. On the other hand weaker relaxations are computed faster so they let us do more iterations and explore much greater portion of the problem tree. The greater number of iterations increases chances for guessing the optimal solution as well as obtaining good lower bounds for problems that are small enough.

Appendix A

A.1 Schur complement

Theorem A.1 (Schur complement lemma for PSD [10, 5])**.** *Let M be a symmetric matrix of the form*

$$M = \begin{pmatrix} A & B \\ B^T & C \end{pmatrix}.$$

The following conditions are equivalent:

1. $M \succeq 0$ *(M is positive semidefinite).*
2. $A \succeq 0, \quad (I - AA^\dagger)B = 0, \quad C - B^T A^\dagger B \succeq 0.$
3. $A \succeq 0, \quad \mathcal{R}(B) \subseteq \mathcal{R}(A), \quad C - B^T A^\dagger B \succeq 0.$

(The roles of A and C can be switched.) Where $A^\dagger$ denotes pseudoinverse of A and $\mathcal{R}(B)$ denotes column range of B.

Proof. (will be added later)

A.2 Cones

This Section is focused on providing introduction to theory of cones and overview of basic claims and definitions. Majority of the content in this section is from [8]. In the end of this Section we provide examples dealing with cones $\mathbb{R}^n_+$, $\mathbb{Q}^n_+$, $\mathbb{S}^n_+$, $\mathbb{P}^n$, $\mathbb{C}^n$, in order to prove all of the claims and properties used throughout this thesis.

Definition A.2 (Cone)**.** Set $\mathcal{K}$ is cone if for all $x \in \mathcal{K}$ and all $\theta \geq 0$, it holds that $\theta x \in \mathcal{K}$.

Definition A.3 (Proper cone)**.** The cone $\mathcal{K} \subseteq \mathbb{R}^n$ is a proper cone if it has following properties

- $\mathcal{K}$ is closed
- $\mathcal{K}$ is convex (for any $\theta_1, \theta_2 \geq 0$ and $x_1, x_2 \in \mathcal{K}$ also $\theta_1 x_1 + \theta_2 x_2 \in \mathcal{K}$)

- $\mathcal{K}$ has nonempty interior (*int* $\mathcal{K} \neq \emptyset$)
- $\mathcal{K}$ is pointed (does not contain whole line, i.e. if $\pm x \in \mathcal{K}$, then $x = 0$).

Definition A.4 (Dual cone)**.** For any cone $\mathcal{K}$ we define its dual cone $\mathcal{K}^*$ as a set

$$\mathcal{K}^* = \{z \mid \forall x \in \mathcal{K},\ x^T z \geq 0\}. \tag{A.5}$$

If $\mathcal{K} = \mathcal{K}^*$ we say cone $\mathcal{K}$ is selfdual.

In the following section we will need to use the Separating hyperplane theorem.

Theorem A.6 (Separating hyperplane [5])**.** *Let $C \subseteq \mathbb{R}^n$ be a closed convex set and let $x_0 \notin C$. Then there exists $a \in \mathbb{R}^n$, $a \neq 0$ and $b \in \mathbb{R}$, such that $a^T x \geq b$ $\forall x \in C$ and $a^T x_0 < b$.*

Proof. Since this theorem is well known, we only refer to proof in the literature see i.e. [5]. □

Proposition A.7 ([8])**.** *Following holds for dual cone $\mathcal{K}^*$*

1. *$\mathcal{K}^*$ is convex,*
2. *$\mathcal{K}^*$ is closed,*
3. *if int $\mathcal{K} \neq \emptyset$ then $\mathcal{K}^*$ is pointed ,*
4. *if closure of $\mathcal{K}$ is pointed cone then int $\mathcal{K}^* \neq \emptyset$.*

Corollary A.8. *If $\mathcal{K}$ is proper cone then also the dual cone $\mathcal{K}^*$ is proper.*

Proof.

1. Let $z_1, z_2 \in \mathcal{K}^*$. Then from definition of dual cone, for any $\theta_1, \theta_2 \geq 0$ and $x \in \mathcal{K}$ it holds that

$$x^T(\theta_1 z_1 + \theta_2 z_2) = \theta_1 x^T z_1 + \theta_2 x^T z_2 \geq 0,$$

which implies convexity of $\mathcal{K}^*$.

2. Let $\{z_n\}_{n=1}^{\infty}$ be a convergent sequence of points in $\mathcal{K}^*$, such that $z_n \to z$. It is enough to show that $z \in \mathcal{K}^*$. Indeed, for any $x \in \mathcal{K}$ and positive integer n we have $x^T z_n \geq 0$, therefore $\lim_{n\to\infty} x^T z_n = x^T z \geq 0$. Hence $z \in \mathcal{K}^*$.

3. Let *int* $\mathcal{K} \neq \emptyset$. It is enough to prove that $\mathcal{K}^* \cup (-\mathcal{K}*) = \{0\}$. Let $y \in \mathcal{K}^* \cup (-\mathcal{K}*)$, then $x^T y = 0$ for all $x \in \mathcal{K}$. Since *int* $\mathcal{K} \neq \emptyset$, there exist open ball $B \subset \mathcal{K}$. For contradiction suppose that $y \neq 0$, then there is $u \in \mathbb{R}^n$ such that $y^T u > 0$. Now take any $x \in B$ and $\alpha > 0$ such that $x + \alpha u \in B$, we have

$$0 = y^T(x + \alpha u) = y^T x + \alpha y^T u = 0 + \alpha y^T u > 0,$$

and that is a contradiction.

4. Denote $\bar{\mathcal{K}}$ the closure of $\mathcal{K}$ and let $\bar{\mathcal{K}}$ be pointed, i.e. $\bar{\mathcal{K}} \cap \left(-\bar{\mathcal{K}}\right) = \{0\}$. First, we will show that $0 \notin conv(S^n \cup \bar{\mathcal{K}})$. By contradiction, if $0 \in conv(S^n \cup \bar{\mathcal{K}})$ then there are some $x^1, \ldots, x^m \in \left(S^n \cup \bar{\mathcal{K}}\right)$ and $theta_1, \ldots \theta_m \geq 0$, such that $\sum_{i=1}^m = 1$ and $0 = \sum_{i=1}^m \theta_i x^i$. Let j be an index for which $\theta_j \neq 0$ (there must be at least one such index). Then

$$\theta_j x^j \in \bar{\mathcal{K}} \quad \text{and} \quad -\theta_j x^j = \sum_{i \neq j} \theta_i x^i \in \bar{\mathcal{K}}.$$

This is a contradiction, because $\bar{\mathcal{K}}$ is pointed, hence $0 \notin conv(S^n \cup \bar{\mathcal{K}})$. Since $conv(S^n \cup \bar{\mathcal{K}})$ is closed and convex, by separating hyperplane theorem A.6 there exists nonzero $a \in \mathbb{R}^n$ and $b \in \mathbb{R}$ such that $a^T x \geq b$ for any $x \in (S^n \cup \bar{\mathcal{K}})$ and $a^T 0 < b$. Therefore $b > 0$ and for any nonzero $y \in \bar{\mathcal{K}}$, since the $y/\|y\| \in (S^n \cup \bar{\mathcal{K}})$ we have

$$a^T y = \|y\| a^T \frac{y}{\|y\|} \geq \|y\| b > 0,$$

hence $a \in \mathcal{K}^*$.

We will show that the ball with center a and radius b ($B(a, b)$) is contained in $\mathcal{K}^*$, hence $a \in int\ \mathcal{K}^*$. Take arbitrary $u \in \mathbb{R}^n$, such that $\|u\| < b$. From Cauchy-Schwartz inequality we have

$$u^T y \geq -|u^T y| \geq -\|u\|\|y\|, \quad \text{i.e.} \quad u^T y/\|y\| \geq -\|u\|.$$

Now

$$(a+u)^T y = a^T y + u^T y \geq b\|y\| - \|u\|\|y\| = (b - \|u\|)\|y\| \geq 0.$$

Therefore $a \in int\ \mathcal{K}^* \neq \emptyset$ and proof is complete.

□

Proposition A.9. *If $\mathcal{K}$ is closed convex cone then $\mathcal{K}^{**} = \mathcal{K}$.*

Proof. Recall that

$$z \in \mathcal{K}^{**} \;\Leftrightarrow\; y^T z \geq 0 \;\forall y \in \mathcal{K}^*.$$

The inclusion $\mathcal{K} \subseteq \mathcal{K}^{**}$ follows directly from the definition of $\mathcal{K}^*$, since $y \in \mathcal{K}^*$ if and only if $x^T y \geq 0$ for all $x \in \mathcal{K}$.

Next we will show the second inclusion $\mathcal{K}^{**} \subseteq \mathcal{K}$ by contradiction. Suppose that there is $z \in \mathcal{K}^{**}$ with $z \notin \mathcal{K}$. Since $\mathcal{K}$ is closed and convex, there is a hyperplane separating z and $\mathcal{K}$ (see Theorem A.6), i.e. there is an $a \in \mathbb{R}^n$, such that

$$a^T z < 0 \quad \text{and} \quad a^T x \geq 0, \; \forall x \in \mathcal{K}.$$

The second inequality implies that $a \in \mathcal{K}^*$, but then $a^T z < 0$ contradicts $z \in \mathcal{K}^{**}$.

All in all, we have $\mathcal{K}^{**} = \mathcal{K}$. □

Proposition A.10 ([8])**.** *Let $\mathcal{K} = \mathcal{K}_1 \times \cdots \times \mathcal{K}_m$ be a Cartesian product of cones. Then*

1. *$\mathcal{K}$ is a cone.*

2. *If $\mathcal{K}_1, \cdots, \mathcal{K}_m$ are convex cones, then $\mathcal{K}$ is convex.*

3. *If $\mathcal{K}_1, \cdots, \mathcal{K}_m$ are selfdual cones, then $\mathcal{K}$ is selfdual.*

4. *If $\mathcal{K}_1, \cdots, \mathcal{K}_m$ are proper cones, then $\mathcal{K}$ is proper.*

Proof. 8 □

TODO:

- *carthesian product of proper/self-dual cones*
- $\mathbb{R}^n_+$, $\mathbb{Q}^n$, $\mathbb{S}^n_+$ *are proper, self-dual cones + figures.*
- *example of converting SOCP to SDP*
- *example of converting QP to SOCP*
- *Moore-Penrose pseudoinverse*

A.2.1 Examples

Example A.11 (Closed convex cones)**.** The following sets are closed convex cones

1. Nonnegative orthant $\mathbb{R}^n_+$,

2. Second order cone

$$\mathbb{Q}^n = \{x \in \mathbb{R}^n \mid x = (x_0, \bar{x}) \in \mathbb{R} \times \mathbb{R}^{n-1}, \|\bar{x}\|_2 \leq x_0\},$$

3. Set of symmetric positive semidefinite matrices

$$\mathbb{S}^n_+ = \{M \in \mathbb{S}^n \mid x^T M x \geq 0, \ \forall x \in \mathbb{R}^n\},$$

4. Set of copositive matrices

$$\mathbb{C}^n = \{M \in \mathbb{S}^n \mid x^T M x \geq 0, \ \forall x \in \mathbb{R}^n_+\},$$

5. Set of completely positive matrices

$$\mathbb{P}^n = \{M \in \mathbb{S}^n \mid M = \sum_{i=1}^{l} x^i x^{iT} \ \text{ where } x^i \in \mathbb{R}^n_+ \ (i = 1, \ldots, n)\}.$$

Proof version 1. All the parts follow straightforward from the definition. □

Proof version 2. The fact that each of the sets is closed is simple since all of them are defined only using nonstrict inequalities (and symmetry), which will also hold in limit.

1. Follows simply from the definition.
2. (It is a cone). Let $(x_0, \bar{x}) = x \in \mathbb{Q}^n$ and $\theta \in \mathbb{R}^n_+$. Then for $\theta x = (\theta x_0, \theta\bar{x})$ we have
$$\|\theta\bar{x}\|_2 = \theta\|\bar{x}\|_2 \leq \theta x_0.$$
(It is convex). Let $x = (x_0, \bar{x}), y = (y_0, \bar{y}) \in \mathbb{Q}^n$ and $\theta_1, \theta_2 \geq 0$. Then
$$\begin{aligned} \theta_1 x + \theta_2 y &= (\theta_1 x_0 + \theta_2 y_0,\ \theta_1\bar{x} + \theta_2\bar{y}), \\ \|\theta_1\bar{x} + \theta_2\bar{y}\|_2 &\leq \|\theta_1\bar{x}\|_2 + \|\theta_2\bar{y}\|_2 \leq \theta_1 x_0 + \theta_2 y_0. \end{aligned}$$
3. (It is a cone) Let $M \in \mathbb{S}^n_+$, that is $x^T M x \geq 0\ \forall x \in \mathbb{R}^n$ and $\theta \geq 0$. Then
$$x^T(\theta M)x = \theta(x^T M x) \geq 0\ \forall x \in \mathbb{R}^n \ \Leftrightarrow\ (\theta M) \in \mathbb{S}^n_+.$$
(It is convex) Let $M_1, M_2 \in \mathbb{S}^n_+$, and $\theta_1, \theta_2 \geq 0$. Then
$$x^T(\theta_1 M_1 + \theta_2 M_2)x = x^T\theta_1 M_1 x + x^T\theta_2 M_2 x \geq 0\,\forall x \in \mathbb{R}^n \ \Leftrightarrow\ (\theta_1 M_1 + \theta_2 M_2) \in \mathbb{S}^n_+.$$
4. Analogically as for $\mathbb{S}^n_+$, only $\mathbb{R}^n$ is replaced by $\mathbb{R}^n_+$.
5. (It is a cone) Let $M \in \mathbb{P}^n$ and $\theta \geq 0$. Since M is completely positive, there are nonnegative vectors x^i, $i = 1, \ldots, l$ such that
$$\theta M = \theta \sum_{i=1}^{l} x^i x^{iT} = \sum_{i=1}^{l} (\sqrt{\theta}x^i)(\sqrt{\theta}x^i)^T.$$
(It is convex) Let $M_1, M_2 \in \mathbb{P}^n$ and $\theta_1, \theta_2 \geq 0$. Since M_1, M_2 are completely positive, there are nonnegative vectors x^i, y^j, ($i = 1, \ldots, l_1$ and $j = 1, \ldots, l_2$) such that
$$\begin{aligned} \theta_1 M_1 + \theta_2 M_2 &= \theta_1 \sum_{i=1}^{l_1} x^i x^{iT} + \theta_2 \sum_{j=1}^{l_2} y^j y^{jT} \\ &= \sum_{i=1}^{l_1} (\sqrt{\theta_1}x^i)(\sqrt{\theta_1}x^i)^T + \sum_{j=1}^{l_2} (\sqrt{\theta_2}y^i)(\sqrt{\theta_2}y^i)^T. \end{aligned}$$

□

Example A.12 (Proper cones)**.** Following cones are proper

1. $\mathbb{R}^n_+$, 2. $\mathbb{Q}^n$, 3. $\mathbb{S}^n_+$, 4. $\mathbb{C}^n$, 5. $\mathbb{P}^n$.

Proof. Since we have shown that all of them are closed convex cones (see Example A.11), we only need to prove that they are pointed and have nonempty interiors. The first two cases are obviously pointed and have interior points (for example all ones vector for $\mathbb{R}^n_+$ and $(2, 0^T)$ for $\mathbb{Q}^n$ are interior points - an open unit ball with these centres belongs to these cones).

3. If positive semidefinite matrix M is nonzero, it has at leas one positive eigenvalue $\lambda > 0$ and corresponding eigenvector u such that $u^T M u > 0$. But then $u^T(-M)u = -u^T M u < 0$, so $-M$ is not in $\mathbb{S}^n_+$ and therefore it is pointed. Any positive definite matrix is an interior point of $\mathbb{S}^n_+$, so it has nonempty interior.

4. The $\mathbb{C}^n$ is closed, convex cone (see Example A.11). It is also easy to see that $\mathbb{C}^n$ has nonempty interior, since it contains $\mathbb{S}^n_+$ (which is proper).

   We only need to show that $\mathbb{C}^n$ is pointed. For $M \neq 0$, $M \in \mathbb{C}^n \cap (-\mathbb{C}^n)$, then $x^T M x \geq 0$ and $x^T(-M)x \geq 0$ for all $x \in \mathbb{R}^n_+$, implying that $x^T M x = 0$ for all $x \in \mathbb{R}^n_+$. Since any vector $v \in \mathbb{R}^n$ can be written as $v = x - y$ for some $x, y \in \mathbb{R}^n_+$ it also holds that $v^T M v = 0$ for all $v \in \mathbb{R}^n$, hence $M = 0$.

   Therefore $\mathbb{C}^n$ is a proper cone.

5. From the Corrolary A.8 since $\mathbb{C}^n$ is a proper cone, its dual $\mathbb{P}^n$ is also proper.

□

Example A.13 (Dual cones)**.** The following properties about dual cones holds

1. $\mathbb{R}^n_+$ is selfdual cone,

2. $\mathbb{Q}^n$ is selfdual cone,

3. $\mathbb{S}^n_+$ is selfdual cone,

4. $(\mathbb{C}^n)^* = \mathbb{P}^n$ and $(\mathbb{P}^n)^* = \mathbb{C}^n$.

Proof. 1. This is an easy observation.

2. Recall that $y \in (\mathbb{Q}^n)^*$ if and only if $x^T y \geq 0$ for all $x \in \mathbb{Q}^n$. First we will show that $\mathbb{Q}^n \subseteq (\mathbb{Q}^n)^*$. For any $x = (x_0, \bar{x}), y = (y_0, \bar{y}) \in \mathbb{Q}^n$ it holds that

$$x^T y = x_0 y_0 + \bar{x}^T \bar{y} \geq \|\bar{x}\|_2 \|\bar{y}\|_2 - |\bar{x}^T \bar{y}| \geq 0,$$

   where the first inequality holds from $x_0 \geq \|\bar{x}\|_2$, $y_0 \geq \|\bar{y}\|_2$ and the second inequality follows from Cauchy-Schwartz inequality.

   Now $(\mathbb{Q}^n)^* \subseteq \mathbb{Q}^n$. Let $y \notin \mathbb{Q}^n$, we will show that $y \notin (\mathbb{Q}^n)^*$. For such $y = (y_0, \bar{y})$ it holds that $y_0 < \|\bar{y}\|_2$. Let $x = (\|\bar{y}\|_2, -\bar{y})$, this x belongs to $\mathbb{Q}^n$, but $x^T y = \|\bar{y}\|_2 (y_0 - \|\bar{y}\|_2) < 0$. So $y \notin (\mathbb{Q}^n)^*$ and we are done.

3. Recall that $M \in (\mathbb{S}^n_+)^*$ if and only if $P \bullet M \geq 0$ for all $P \in \mathbb{S}^n_+$. First we will first show that $\mathbb{S}^n_+ \subseteq (\mathbb{S}^n_+)^*$. For any $M, P \in \mathbb{S}^n_+$ it holds that

$$P \bullet M = Tr(PM) = \sum_{i=1}^{n} \lambda_i u_i^T M u_i \geq 0,$$

   where $\lambda_i \geq 0$, u_i $(i = 1, \ldots, n)$, are eigenvalues and eigenvectors of P.

   Not the other inclusion. Let $M \notin \mathbb{S}^n_+$ be a symmetric $n \times n$ matrix, we will show that $M \notin (\mathbb{S}^n_+)^*$. Since M is not positive semidefinite, it has a negative eigenvalue λ and corresponding eigenvector u. Then matrix $uu^T \in \mathbb{S}^n_+$ excludes M from $(\mathbb{S}^n_+)^*$ since $uu^T \bullet M = u^T M u = \lambda < 0$.

Therefore $\mathbb{S}^n_+ = (\mathbb{S}^n_+)^*$

4. First we will show $(\mathbb{P}^n)^* = \mathbb{C}^n$. From the definition, using the inner product $Tr(M^T P) = M \bullet P = svec(M)^T svec(P)$ we have

$$(\mathbb{P}^n)^* = \{M \in \mathbb{S}^n \mid M \bullet P \geq 0,\ \forall P \in \mathbb{P}^n\}.$$

Observe that for any $C \in \mathbb{C}^n$ and $P =\in \mathbb{P}^n$, $P = \sum x^i x^{iT}$ it holds that

$$C \bullet P = \sum C \bullet x^i x^{iT} = \sum x^{iT} C x^i \geq 0.$$

This implies $\mathbb{C}^n \subseteq (\mathbb{P}^n)^*$.

To prove the other inclusion $(\mathbb{P}^n)^* \subseteq \mathbb{C}^n$, take any $M \notin \mathbb{C}^n$. We will show that M is neither in $(\mathbb{P}^n)^*$. For such M, there is a $x \in \mathbb{R}^n_+$, such that $x^T M x < 0$. Indeed, the $xx^T \in \mathbb{P}^n$ then excludes M from $(\mathbb{P}^n)^*$ since $M \bullet xx^T = x^T M x < 0$.

We have proven that $(\mathbb{P}^n)^* = \mathbb{C}^n$. Since $\mathbb{P}^n$ is a closed convex cone, it holds that $\mathbb{P}^n = (\mathbb{P}^n)^{**} = ((\mathbb{P}^n)^*)^* = (\mathbb{C}^n)^*$, and the proof is complete.

□

References

TODO: Unified refeference (same formats and orderings for source, year, pages, volume, journal names)

[1] S. Burer, S.Kim, M. Kojima, *Faster, but Weaker, Relaxations for Quadratically Constrained Quadratic Programs,* April, 2013
http://www.is.titech.ac.jp/~kojima/articles/socpsdp.pdf

[2] S. Kim, M. Kojima, *Exact Solutions of Some Nonconvex Quadratic Optimization Problems via SDP and SOCP Relaxations,*January 2002.Computational Optimization and Applications Vol.26 (2) 143-154 (2003)
http://www.is.titech.ac.jp/research/research-report/B/B-375.ps.gz

[3] S. Kim, M. Kojima, M. Yamashita, *Second Order Cone Programming Relaxation of a Positive Semidefinite Constraint,* July 2002. Optimization Methods and Software Vol.18 (5) 535-541 (2003).
http://www.is.titech.ac.jp/~kojima/articles/B-381.pdf

[4] S. Kim, M. Kojima, *Second Order Cone Programming Relaxations of Nonconvex Quadratic Optimization Problems,* Optim. Methods Softw., 15(3-4):201-224, 2001
http://www.researchgate.net/publication/2637515_Second_Order_Cone_Programming_Relaxation_of_Nonconvex_Quadratic_Optimization_Problems

[5] S. Boyd, L. Vandenberghe, *Convex Optimization,* Cambridge University Press 2004, seventh printing with corrections 2009
http://web.stanford.edu/~boyd/cvxbook/bv_cvxbook.pdf

[6] R. M. Freund, *Introduction to Semidefinite Programming (SDP),* 2009
http://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-251j-introduction-to-mathematical-programming-fall-2009/readings/MIT6_251JF09_SDP.pdf

[7] F. Alizadeh, D. Goldfarb, *Second-order cone programming,* Springer, 2002

[8] K. Pokorná, *Second order cone programing, (Master's thesis),* 2013
http://www.iam.fmph.uniba.sk/studium/efm/diplomovky/2013/pokorna/diplomovka.pdf

[9] M. S. Lobo, L. Vandenberghe, S. Boyd, H. Lebret, *Applications of second-Order cone programming,* Linear Algebra and its Applications 284, 1998, 193-228
`http://www.seas.ucla.edu/~vandenbe/publications/socp.pdf`

[10] J. Gallier, *The Schur Complement and Symmetric Positive Semidefinite (and Definite) Matrices,* December 10, 2010
`http://www.cis.upenn.edu/~jean/schur-comp.pdf`

[11] R. Grone, C. Johnson, E. Sá, H. Wolkowicz, *Positive definite completions of partial hermitian matrices,* Linear Algebra Applications, 58:109-124, 1984.

[12] H. Wolkowicz, R. Saigal,L. Vandenberghe, *Handbook of Semidefinite Programming: Theory, Algorithms, and Applications,* ISBN 978-1-4615-4381-7, second printing 2003

[13] B. Kocuk, S. S. Dey, X. A. Sun, *Inexactness of SDP Relaxation and Valid Inequalities for Optimal Power Flow,* IEEE Transactions on Power Systems, 2014

[14] H. Dong, J. Linderoth, *On Valid Inequalities for Quadratic Programming with Continuous Variables and Binary Indicators* Springer Volume 7801 of the series Lecture Notes in Computer Science pp 169-180,

[15] K. M. Anstreicher, *Semidefinite Programming versus the Reformulation Linearization Technique for Nonconvex Quadratically Constrained Quadratic Programming* Optimization Online, May 2007
`http://www.optimization-online.org/DBHTML/2007/05/1655.html`

[16] A. Qualizza , P. Belotti, F. Margot, *Linear Programming Relaxations of Quadratically Constrained Quadratic Programs* IMA Volume Series 2010
`http://wpweb2.tepper.cmu.edu/fmargot/PDF/sdp.pdf`

[17] H. D. Sherali, W. P. Adams, *A reformulation-linearization technique for solving discrete and continuous nonconvex problems* Kluwer, Dordrecht 1998

[18] H.D. Sherali, B.M.P. Fraticelli, *Enhancing RLT relaxations via a new class of semidefinite cuts* J. Global Optim. 22 (2002), 233-261

[19] X. Zheng, X. Sun, D. Li, *Nonconvex quadratically constrained quadratic programming: best d.c. decompositions and their sdp representations* Journal of Global Optimization, 50:695-712, 2011

[20] N. Arima, S. Kim, M. Kojima, *A quadratically constrained quadratic optimization model for completely positive cone programming.* SIAM J. Optim., 23(4):2320-2340, 2013.

[21] S. Burer *On the copositive representation of binary and continuous non- convex quadratic programs.* Mathematical Programming, 120, 479-495, 2009.

[22] G. Eichfelder, J. Povh *On the set-semidefinite representation of nonconvex quadratic programs over arbitrary feasible sets,* Optim. Lett., DOI 10.1007/s11590-012-0450-3, 2012.

[23] S. Kim, M. Kojima, K. Toh, *A Lagrangian-DNN Relaxation: a Fast Method for Computing Tight Lower Bounds for a Class of Quadratic Optimization Problems,* Mathematical Programming, 156:161-187, 2016.

[24] N. Arima, S. Kim, M. Kojima, K. Toh, *A Robust Lagrangian-DNN Method for a Class of Quadratic Optimization Problems* Research Report B-472, Department of Mathematical and Computing Sciences, Tokyo Institute of Technology, Oh-Okayama, Meguro-ku, Tokyo 152-8552, February 2016.

[25] M. Dur, *Copositive Programming – a Survey* Recent Advances in Optimization and its Applications in Engineering, pp 3-20, August 2010

[26] K. G. Murty, S. N. Kabadi, *Some NP-complete problems in quadratic and non-linear programming* Math. Program., 39, 117–129, 1987.

[27] E. Balas, S. Ceria, G. Cornuéjols, *A lift-and-project cutting plane algorithm for mixed 0-1 programs,* Math. Program., 58:295–324, 1993.

[28] L. Lovász, A. Schrijver, *Cones of matrices and set-functions and 0-1 optimization,* SIAM Journal on Optimization, 1:166–190, 1991

[29] H. Sherali, W. Adams. *A hierarchy of relaxation between the continuous and convex hull representations,* SIAM J. Discret. Math., 3:411–430, May 1990.

[30] J. Lasserre. *An explicit exact SDP relaxation for nonlinear 0-1 programs,* In IPCO, pages 293–303, 2001.

[31] J. Lasserre. *Global optimization with polynomials and the problem of moments,* SIAM Journal on Optimization, 11(3):796–817, 2001.

[32] M. Laurent, *A Comparison of the Sherali-Adams, Lovász-Schrijver and Lasserre Relaxations for 0 - 1 Programming* Mathematics of Operations Research, 28:470-496, 2001

[33] T. Rothvoß, *The Lasserre hierarchy in Approximation algorithms* Lecture Notes for the MAPSP 2013 Tutorial, available online `https://www.math.washington.edu/~rothvoss/lecturenotes/lasserresurvey.pdf`

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

[35] U. Raber, *A Simplicial Branch-and-Bound Method for Solving Nonconvex All-Quadratic Programs* Journal of Global Optimization 13: 417–432, 1998

[36] M. Armbruster, M. Fügenschuh, Ch. Helmberg, A. Martin, *LP and SDP branch-and-cut algorithms for the minimum graph bisection problem: a computational comparison* Math. Prog. Comp. (2012) 4:275–306

[37] M. X. Goemans, D. P. Williamson, *Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming* J. ACM, 42(6):1115-1145, 1995

[38] M. Muramatsu, T. Suzuki, *A new second-order cone programming relaxation for MAX-CUT problems,* Journal of the Operations Research Society of Japan 2003, Vol. 46, No. 2, 164-177

[39] P.M. Pardalos, S.A. Vavasis, *Quadratic programming with one negative eigenvalue is NP-hard* J. Global Optim. 1, 15–22, 1991