Faculty of Mathematics, Physics and Informatics
Comenius University, Bratislava

Conic Relaxations

Master's Thesis

Bc. Matúš Stehlík

Bratislava, 2016

Fakulta Matematiky, Fyziky a Informatiky
Univerzita Komenského, Bratislava

Kónické Relaxácie

(Diplomová práca)

Bc. Matúš Stehlík

Študijný program:	Matematika
Študijný odbor:	1113 Matematika
Školiace pracovisko:	Katedra aplikovanej matematiky a štatistiky
Školiteľ:	RNDr. Mária Trnovská, PhD.

Bratislava, 2016

27013866

Univerzita Komenského v Bratislave
Fakulta matematiky, fyziky a informatiky

ZADANIE ZÁVEREČNEJ PRÁCE

Meno a priezvisko študenta:	Bc. Matúš Stehlík
Študijný program:	matematika (Jednoodborové štúdium, magisterský II. st., denná forma)
Študijný odbor:	matematika
Typ záverečnej práce:	diplomová
Jazyk záverečnej práce:	anglický
Sekundárny jazyk:	slovenský

Názov:	Conic relaxations *Kónické relaxácie*
Cieľ:	Skúmanie možnosi relaxácie úloh nelineárnej optimalizácie pomocou kónických úloh, odhadov presnosti, kritérií exaktnosti.

Vedúci:	RNDr. Mária Trnovská, PhD.
Katedra:	FMFI.KAMŠ - Katedra aplikovanej matematiky a štatistiky
Vedúci katedry:	prof. RNDr. Daniel Ševčovič, CSc.
Dátum zadania:	08.01.2015

Dátum schválenia: 20.01.2015

prof. RNDr. Ján Filo, CSc.
garant študijného programu

....
študent

....
vedúci práce

Faculty of Mathematics, Physics and Informatics
Comenius University, Bratislava

Conic Relaxations

(Master's Thesis)

Bc. Matúš Stehlík

Study programme:	Mathematics
Field of study:	1113 Mathematics
Department:	Department of Applied Mathematics and Statistics
Supervisor:	RNDr. Mária Trnovská, PhD.

Bratislava, 2016

27013866

Comenius University in Bratislava
Faculty of Mathematics, Physics and Informatics

THESIS ASSIGNMENT

Name and Surname:	Bc. Matúš Stehlík
Study programme:	Mathematics (Single degree study, master II. deg., full time form)
Field of Study:	Mathematics
Type of Thesis:	Diploma Thesis
Language of Thesis:	English
Secondary language:	Slovak

Title:	Conic relaxations
Aim:	Investigate the possibility of relaxation of the nonlinear optimization problems using the conic programming classes, precision estimates and exactness criteria.

Supervisor:	RNDr. Mária Trnovská, PhD.	
Department:	FMFI.KAMŠ - Department of Applied Mathematics and Statistics	
Head of department:	prof. RNDr. Daniel Ševčovič, CSc.	
Assigned:	08.01.2015	
Approved:	20.01.2015	prof. RNDr. Ján Filo, CSc. Guarantor of Study Programme

....

Student Supervisor

Čestné vyhlásenie
Vyhlasujem, že som túto diplomovú prácu vypracoval samostatne, pod vedením vedúceho práce a uviedol som všetku použitú odbornú literatúru.

V Bratislave dňa 4. 5. 2016

....
Bc. Matúš Stehlík

Poďakovanie

Ďakujem vedúcej tejto diplomovej práce RNDr. Márii Trnovskej PhD. za jej cenné rady, postrehy, odborné vedenie a pomoc počas vypracovávania práce.

Abstrakt

STEHLÍK, Matúš: *Kónické relaxácie.* Diplomová práca. Univerzita Komenského v Bratislave, Fakulta matematiky, fyziky a informatiky, Katedra matematickej analýzy a numerickej matematiky. Školiteľ: RNDr. Mária Trnovská PhD., Bratislava, 2016, … strán.

Diplomová práca sa venuje možnostiam relaxácie nekonvexných kvadratických optimalizačných úloh pomocou tried kónického programovania. Práca ponúka prehľad metód pre vytváranie kónických relaxácii. Podrobne sa venuje ich aplikácii na problém maximálneho rezu v grafe a porovnaniu ich efektivity na pri riešení tohto problému.

Prvá kapitola uvádza triedy kónického programovania a vzťahy medzi nimi. V druhej kapitole sú zhrnuté metódy relaxovania pomocou uvedených tried. Tretia kapitola rozoberá aplikácie relaxačných techník na problém maximálneho rezu, kotré sú porovnané pomocou numerických experimentov v štvrtej kapitole. Dodatok uvádza základy teórie kužeľov a na príkladoch ukazuje vlastnosti kužeľov používaných v práci.

Kľúčové slová: relaxácia, kužeľ, kónické programovanie, optimalizácia, semidefinitné programovanie, platé nerovnosti, problém maximálneho rezu

Abstract

STEHLÍK, Matúš: *Conic relaxations.* Master thesis. Comenius University in Bratislava, Faculty of mathematics, physics and informatics, Department of mathematical analysis and numerical mathematics. Advisor: RNDr. Mária Trnovská PhD., Bratislava, 2016, … pages.

This master's thesis explores the possibility of relaxation of the nonlinear quadratic optimization problems using the conic programming classes. The thesis offers a survey of techniques used for constructing conic relaxations. Explores in detail their applications to the maximum cut problem and compares their performance on the instances of this problem.

First chapter introduces conic optimization classes and the relations between them. Second chapter surveys the methods of relaxation with conic programs. Third chapter analyses application of the relaxation techniques to the maximum cut probem, which are numerically compared in the fourth chapter. The appendix covers basic theory of cones with examples showing the properties of the cones used throughout the thesis.

Key words: relaxation, cone, conic programming, optimization, semidefinite programming, valid inequalities, maximum cut problem

Contents

Introduction **1**

Notation . . . 4

1 Conic optimization classes **5**

1.1 Linear programming . . . 5

1.2 Second order cone programing . . . 6

1.2.1 Relation to previous classes . . . 6

1.3 Semidefinite programming . . . 8

1.3.1 Relation to previous classes . . . 9

1.4 Copositive and completely positive programming . . . 9

1.4.1 Relation to previous classes . . . 10

1.5 General conic programming approach . . . 13

1.6 Dual problems . . . 14

1.6.1 Dual problem of conic programming . . . 14

1.6.2 Dual problem of QCQP . . . 15

1.6.3 Second dual of QCQP . . . 16

2 Relaxations **19**

2.1 SDP relaxation of QCQP . . . 20

2.2 CQP and SOCP relaxation . . . 22

2.2.1 SOCP relaxation of semidefinite constraint . . . 22

2.2.2 SOCP in the original variable . . . 23

2.2.3 Relaxing the semidefinite constraint and reducing the number of variables . . . 24

2.3 Mixed SOCP-SDP relaxation . . . 25

2.4 LP relaxation of QCQP . . . 27

2.4.1 Reformulation linearization technique (RLT) . . . 28

2.4.2 Positive semidefinite cuts . . . 29

2.5 Strengthening the SDP relaxation . . . 30

2.5.1 SDP + RLT relaxation 31
2.6 Copositive and completely positive programming representation . . . 32
2.6.1 The Lagrangian doubly nonnegative relaxation 32
2.7 Lasserre hierarchy 34

3 The maximum cut problem 37
3.1 Problem formulation 37
3.1.1 Quadratic formulation of the max-cut 38
3.2 Goemans and Williamson analysis 38
3.2.1 Semidefinite relaxation of the max-cut 41
3.3 Relaxations of the max-cut 41
3.3.1 Semidefinite relaxation 41
3.3.2 Triangle inequalities 42
3.3.3 SOCP relaxations 43
3.3.4 Mixed SOCP-SDP relaxation 46
3.3.5 LP relaxations 47
3.3.6 Doubly nonnegative relaxations 49

4 Computational comparison 53
4.1 Single bound comparison 54
4.1.1 Small sized instances 54
4.1.2 Medium sized instances 55
4.1.3 Large scale instances 58
4.1.4 Relation of Mix and SDP 58
4.2 Branch and bound 60
4.2.1 Our branch and bound framework 61
4.2.2 Branch and bound comparison 64

Conclusion 67

A Appendix 69
A.1 Schur complement 69
A.2 Cones 69
A.3 Examples 72

References 77

List of Figures

4.1 Comparison of relaxations - small, sparse instances of max-cut 55
4.2 Comparison of relaxations - small, dense instances of max-cut 1 . . . 56
4.3 Comparison of relaxations - small, dense instances of max-cut 2 . . . 56
4.4 Comparison of relaxations - medium, dense instances of max-cut . . . 57
4.5 Comparison of relaxations - a time-precision trade off 57
4.6 Comparison of relaxations - large, dense instances of max-cut 1 . . . 59
4.7 Comparison of relaxations - large, dense instances of max-cut 2 . . . 59
4.8 Branch and bound comparison of the relaxation methods 62
4.9 Branch and bound for mixed SOCP-SDP relaxation 63
4.10 Branch and bound for SDP relaxation 63

List of Tables

4.1 Branch and bound comparison table 61

Introduction

In this thesis we will study relaxation techniques for the quadratically constrained quadratic programs (QCQP).

Definition 1. The Quadratically Constrained Quadratic Program (QCQP) in the standard form is

$$\begin{array}{ll} \text{minimize} & x^T P_0 x + q_0^T x + r_0 \\ \text{subject to} & x^T P_k x + q_k^T x + r_k \leq 0, \ (k = 1, \ldots, m), \end{array} \tag{2}$$

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.

In the QCQP the objective function as well as the constraints may be nonconvex. The problem has been proved to be NP-hard in general [41], while several special subclasses of QCQP have been identified to be polynomially solvable (see [6]).

However, the QCQP represents many problems with applications. For instance it includes 0-1 programming, which describes various NP-hard problems, such as knapsack, stable set, maximum cut, quadratic assignment problem, etc. Therefore QCQP is worth solving, or rather approximating since it is computationally intractable.

One of the possible approaches is relaxing QCQP with convex problems which can be solved in polynomial time, namely linear programming (LP) [12], second order cone programming (SOCP) [7], or semidefinite programming (SDP) [4].

Since the relaxed problems are not equivalent to the original problem, solving them usually does not provide an exact solution. However, it gives a bound for optimal value. More specifically, in minimization problems, the optimal value of relaxation is a lower bound on the optimal value of the original problem.

There are also methods (e.g. a projection or rounding procedures) to extract a feasible solution of the original problem from the optimum of relaxation. The feasible solution provides the opposite - the upper bound.

The quality of bounds depends widely on the problem and method of relaxation. For some problems the approximation ratio for certain relaxation is guaranteed. A widely known example is the SDP relaxation for the maximum cut by Goemans and WIlliamson [37], which has guaranteed 0.878 approximation bound.

On the other hand, for many combinatorial problems there are also known certain inapproximability bounds which cannot be guaranteed for any polynomial approximation algorithm unless P is NP [40].

Bounds on the optimal value can be used to find the optimal solution with a proof of optimality. Such proof can be done e.g. by the so-called branch and bound (BnB) method, which is based on recursive subdividing of the feasible region into multiple segments and computing bounds on each of them separately. Whenever a bound for a segment is worse than the best known feasible solution, the segment is ruled out from consideration, because it was proven that it does not contain the optimal solution.

The BnB method is usually used in combinatorial optimization, where the variables are discrete [36], however, it can be also adapted for nonconvex continuous optimization when the feasible region is compact [34, 35].

In practice, an important factor is time complexity, hence a relaxation giving the tightest bounds is not always a method of choice. Especially when using the BnB framework, where a sequence of relaxations needs to be solved [36].

Generally, SDP relaxation is supposed to produce stronger bounds then SOCP or LP relaxations, because those are subclasses of semidefinite programming. On the other hand, for the same reason, SOCP and LP relaxations should be easier to solve, therefore shorter computation time is expected advantage of these approaches. In fact, there is a certain speed versus quality "trade-off". An interesting method was proposed by Burer, Kim and Kojima [5] for producing a relaxation "in-between" SDP and SOCP.

It is known that the quality of relaxation also depends on the particular formulation of the problem. Adding redundant constraints to the problem may result in new constraints, called valid inequalities, tightening the relaxation [16, 17, 18]. A popular method used for creating valid inequalities is the reformulation linearization technique (RLT) [13, 14, 15], which is based on multiplying pairs of linear constraints. In fact, Anstreicher [13] has shown that adding RLT constraints improves the SDP relaxation.

Instead of following the heuristic approach of finding 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. The idea is constructing a sequence of relaxations, each one providing better approximation of the problem, for the price of increasing dimension of the relaxed problem. In particular there are procedures by Balas, Ceria, Cornuéjols [26], Lovász, Schrijver [27], Sherali, Adams [28] or Lasserre [29, 30]. The latter one was shown to be superior in the comparison by Laurent [31].

In 2009, completely positive programming (CPP) relaxation for a class of binary quadratic problems was proposed by Burer [20]. The class was extended to a more general class of QOPs by Eichfelder and Povh [21] and by Arima, Kim and Kojima [19]. Theoretically strong results were presented in their papers [19, 20, 21] showing that the exact optimal values of QOPs in their classes coincide with the optimal values of their CPP relaxation problems.

Despite the CCP programs are convex, they are computationally intractable. On the other hand equivalent CPP formulations offer new insights which can lead to improvements of relaxation methods.

A natural way of relaxing CPP is by doubly nonnegative program (DNN). However, 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 [22] authors Kim, Kojima and Toh developed a technique to obtain Lagrangian DNN relaxation primal-dual pair, with single variable in the dual. Furthermore, they proposed a fast algorithm for solving their relaxation based on combination of bisection method and first order methods. Recently (February 2016), Arima, Kim, Kojima and Toh [23] designed an improvement of the algorithm which achieves better robustness and acceleration.

In this thesis we will examine techniques for creating convex relaxations which can be applied to general QCQP or to some nonconvex subclass of QCQP such as binary quadratic programs.

In the first chapter we will introduce convex optimization classes (LP, SOCP, SDP, CP and CCP) and relations between them. To conclude this chapter we will present a general conic programming class which includes all of previously mentioned programs as a special cases.

The second chapter will describe relaxations and relaxation techniques. In the beginning we will propose our own definition to formally capture what is considered as relaxation in this thesis (not exclusively). Next we will discuss options for relaxing the nonconvex QCQP with problems in all conic classes mentioned in the first chapter, including the approaches for construction of valid inequalities. In the end of the chapter we will present a relaxation hierarchy by Lasserre.

The third chapter we will explore the maximum cut problem. As a motivation, we will present the famous result by Goemans and Williamson giving the 0.878 approximation guarantee for SDP relaxation. Next we will apply most of the relaxation methods from chapter 2, to formulate relaxations of max-cut. The relaxations will also include triangle inequalities - the problem specific valid inequalities for max-cut.

In the last chapter we will computationally compare the relaxation methods for the max-cut stated in the chapter 3. We will perform numerical experiments comparing the performance of relaxations on max-cut instances of various size and sparsity. Next we will describe branch and bound framework and use it to compare the performance of chosen methods in this procedure.

In the appendix we will cover basic properties of cones and include examples showing the important properties of the cones used throughout the thesis.

Notation

Sets

$\mathbb{R}$	-	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 $n \times n$ positive semidefinite 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

Operators

$M \succ 0$	-	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 of its entries in $\mathbb{R}^{n(n+1)/2}$
$(x; y)$	-	Column vector $(x^T, y^T)^T$

Abbreviations

PSD	-	Positive Semidefinite
QCQP	-	Quadratically Constrained Quadratic Program
CQP	-	Convex Quadratic Program (a convex QCQP)
LP	-	Linear Program
SOCP	-	Second Order Cone Program
SDP	-	Semidefinite program
CP	-	Copositive program
CPP	-	Completely positive program

Chapter 1

Conic optimization classes

In this section we will introduce basic optimization classes mentioned above, in particular 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.

$$LP \subset CQP \subset SOCP \subset SDP \subset QCQP.$$

As a sources for this chapter we refer to [1,3] as general sources and specifically to SOCP [7, 9, 10], SDP [4, 7], CP and CPP [19, 20, 24].

1.1 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. [1] .

Definition 1.1. The primal–dual pair of linear programs in standard form is

$$\begin{array}{llcll} & \textit{Primal} & & & \textit{Dual} \\ \text{minimize} & c^T x, & & \text{maximize} & b^T y, \\ \text{subject to} & Ax = b, & & \text{subject to} & A^T y + s = c, \\ & x \in \mathbb{R}^n_+, & & & s \in \mathbb{R}^n_+, \end{array} \tag{1.2}$$

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 1.3. 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.

1.2 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 [1] .

Let us first define second order cone.

Definition 1.4 (Second order cone)**.** We say $\mathbb{Q}^n$ is second order cone of dimension n if

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

Definition 1.5 (SOCP)**.** The primal–dual pair of the Second Order Cone Program (SOCP) in the standard form is

$$\begin{array}{llcll} & \textit{Primal} & \quad & & \textit{Dual} \\ \text{minimize} & c^T x, & & \text{maximize} & b^T y, \\ \text{subject to} & Ax = b, & & \text{subject to} & A^T y + s = c, \\ & x \in \mathbb{Q}^n, & & & s \in \mathbb{Q}^n, \end{array} \tag{1.6}$$

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 1.7. 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 1.8. In general, any program of the form

$$\begin{array}{llcll} & \textit{Primal} & \quad & & \textit{Dual} \\ \min & c^{1T}x^1 + \cdots + c^{kT}x^k, & & \max & b^T y, \\ \text{s.t.} & A^1x^1 + \cdots + A^kx^k = b, & & \text{s.t.} & A^{iT}y + s^i = c^i, \\ & x^i \in \mathbb{Q}^{n_i},\ (i = 1, \ldots, k), & & & s^i \in \mathbb{Q}^{n_i},\ (i = 1, \ldots, k), \end{array} \tag{1.9}$$

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,

$$\mathbb{Q} = \mathbb{Q}^{n_1} \times \mathbb{Q}^{n_2} \times \cdots \times \mathbb{Q}^{n_k}, \tag{1.10}$$

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 [9,10]. In order to keep things simple, we will sometimes consider only the standard form stated in the Definition 1.5, but all the details can be also done for this more general form.

1.2.1 Relation to previous classes

Second Order Cone Programming includes convex LP as special case. We will show that SOCP in fact includes CQP as a subclass. We will demonstrate procedure proposed in [7] used to reformulate CQP as SOCP.

Let us have a CQP. In other words, suppose that $n \times n$ matrices P_k, $k = 0, \dots, m$ are positive semidefinite.

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

First of all, rewrite problem equivalently as

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

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

$$x^T P x + q^T x + r \le 0 \tag{1.11}$$

can be transformed into the second order cone constraint. Suppose that $P \ne 0$ 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 (1.11) as

$$(L^T x)^T (L^T x) \le -q^T x - r. \tag{1.12}$$

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

$$w^T w \le \xi\eta, \quad \xi \ge 0 \ \text{ and } \ \eta \ge 0$$

if and only if they satisfy

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

If we take $w = L^T x$, $\xi = 1$ and $\eta = -q^T x - r$, then inequality (1.12) is equivalent to the second order cone constraint

$$\|v\|_2 \le v_0, \quad \text{where} \quad \begin{pmatrix} v_0 \\ v \end{pmatrix} = \begin{pmatrix} 1 - q^T x - r \\ 1 + q^T x + r \\ 2L^T x \end{pmatrix} \in \mathbb{R}^{h+2}. \tag{1.13}$$

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 1.9.

1.3 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 [1] .

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

Definition 1.14. 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 1.15 (SDP)**.** The primal–dual pair of the Semidefinite Program (SDP) in the standard form is

$$\begin{array}{ll} & \textit{Primal} \\ \text{minimize} & A_0 \bullet X, \\ \text{subject to} & A_k \bullet X = b_k, \\ & (k = 1, \dots, m), \\ & X \in \mathbb{S}^n_+, \end{array} \qquad \begin{array}{ll} & \textit{Dual} \\ \text{maximize} & b^T y, \\ \text{subject to} & \sum_{k=1}^m y_k A_k + S = A_0, \\ & S \in \mathbb{S}^n_+, \end{array} \tag{1.16}$$

where $X \in \mathbb{S}^n_+$, $y = (y_1, \dots, y_m)^T \in \mathbb{R}^m$ and $S \in \mathbb{S}^n$ are the variables; and symmetric matrices $A_0, A_1, \dots, A_m \in \mathbb{S}^n$ and scalars $b_1, \dots, b_m \in \mathbb{R}$ 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 1.17. We define operator $svec : \mathbb{S}^n \to \mathbb{R}^{n(n+1)/2}$, such that for any $n \times n$ symmetric matrix M

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

where

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

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

$$A \bullet X = svec(A)^T svec(X),$$

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.

$$\begin{array}{ll} \text{minimize} & svec(A_0)^T svec(X), \\ \text{subject to} & svec(A_k)^T svec(X) = b_k, \;\; (k = 1, \dots, m), \\ & svec(X) \in \mathcal{K}(\mathbb{S}^n_+), \end{array} \tag{1.18}$$

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 (1.16).

1.3.1 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 (1.18), 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

$$x \in \mathbb{Q}^n \;\Leftrightarrow\; ||\bar{x}|| \leq x_1 \;\Leftrightarrow \left\{ \begin{array}{c} \bar{x}^T\bar{x} \leq x_1^2 \\ 0 \leq x_1 \end{array} \right\}$$
$$\Leftrightarrow \left\{ \begin{array}{c} \frac{\bar{x}^T\bar{x}}{x_1} \leq x_1 \\ 0 \leq x_1 \end{array} \right\} \Leftrightarrow \begin{pmatrix} x_1 & \bar{x}^T \\ \bar{x} & x_1 I_{n-1} \end{pmatrix} \succeq 0.$$

Where last equivalence is provided by Schur complement lemma (see appendix, Theorem A.2). The case when $x_1 = 0$ is trivial, one can see that as zero vector belongs to $\mathbb{Q}^n$ also zero matrix satisfies given semidefinite constraint.

In case of more general standard form of SOCP (1.9), the $x \in \mathbb{Q}$ constraint can be transformed similarly.

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

where M is a block diagonal matrix, with blocks M_i of the form (1.19), for $i = 1, \dots, k$, corresponding to the constraints $x^i \in \mathbb{Q}^{n_i}$.

1.4 Copositive and completely positive programming

The copositive programming (CP) and completely positive programming (CPP) are convex conic classes which are numerically intractable. In fact, it is known that determining whether a matrix is copositive is co-NP-complete [25].

However, these optimization classes 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 [19, 20, 21]. While restating of the problem as an optimization problem over one of these cones does not resolve the difficulty of that problem, 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 computational improvements. (see the survey of Copositive programming [24])

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

Definition 1.19 (Copositive cone)**.** We say $\mathbb{C}^n$ is copositive cone of dimension n if

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

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

Definition 1.20 (Completely positive cone)**.** We say $\mathbb{P}^n$ is completely positive cone of dimension n if

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

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.9 , A.11 in the Appendix).

Definition 1.21. The primal–dual pair of the copositive (CP) and the completely positive (CPP) program is

$$\begin{array}{llcll} & \textit{Primal} & & & \textit{Dual} \\ \text{minimize} & A_0 \bullet X, & & \text{maximize} & b^T y, \\ \text{subject to} & A_k \bullet X = b_k, & & \text{subject to} & \sum_{k=1}^m y_k A_k + S = A_0, \\ & (k = 1, \ldots, m), & & & S \in \mathbb{P}^n, \\ & X \in \mathbb{C}^n, & & & \end{array} \tag{1.22}$$

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 (1.18), 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.

1.4.1 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 two other important cones

Definition 1.23. We will denote

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

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.

Lemma 1.24. *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

$$y^TMy = \sum_i y^Tx^i(x^i)^Ty = \sum_i (y^Tx^i)^2 \geq 0.$$

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 $x^TMx \geq 0$ for all $x \in \mathbb{R}^n$, therefore also for all $x \in \mathbb{R}^n_+$. Hence $\mathbb{S}^n_+ \subset \mathbb{C}^n$. □

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

Proposition 1.25. *It holds that*

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

Remark 1.26. For $n \leq 4$ it hods that $\mathbb{P}^n = \mathbb{S}^n_+ \cap \mathbb{N}^n$ and $\mathbb{S}^n_+ + \mathbb{N}^n = \mathbb{C}^n$.

Proof of Proposition 1.25. 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.9 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

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

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 [20]. The class was extended to a more general class of QOPs by Eichfelder and Povh [21] and by Arima, Kim and Kojima [19]. Theoretically strong results were presented in their papers [19, 20, 21]

Copositive programming is closely related to quadratic and combinatorial optimization. We illustrate this connection by means of the standard quadratic problem (an example from [24]).

$$\begin{array}{ll} \text{minimize} & x^T Q x, \\ \text{subject to} & e^T x = 1, \\ & x \in \mathbb{R}^n_+, \end{array} \tag{1.27}$$

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 $ee^T \bullet x x^T = 1$, one can see that

$$\begin{array}{ll} \text{minimize} & Q \bullet X, \\ \text{subject to} & ee^T \bullet X = 1, \\ & X \in \mathbb{P}^n, \end{array} \tag{1.28}$$

is a relaxation of (1.27). 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 1.29 bellow). Together, these results imply that (1.28) is in fact an exact reformulation of (1.27).

Lemma 1.29. *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 $X \in conv(\Gamma_1)$.

We have shown that $\{X \in \mathbb{P}^n \mid ee^T \bullet X = 1\} \subseteq conv(\Gamma)$. Since the other inclusion is obvious, the proof is complete.

□

1.5 General conic programming approach

All of the previously 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 1.30 (Conic Programming)**.** The primal–dual pair of the Linear Conic Program in the standard form is

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

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.

Remark 1.31. We say $\mathcal{K}$ is a proper cone if it is a cone and is closed, convex, pointed and has nonempty interior (see the Definition A.4 and the Example A.10 in the Appendix). The

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

denotes the dual cone of $\mathcal{K}$ (see the Example A.11 in the Appendix).

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

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

where the variable $x = (x^1, \ldots, x^k)^T$ is the Cartesian product of the variables x^i, constrained by various proper cone constraints, i.e. 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: nonegative orthant $\mathbb{R}^n_+$, second order cone $\mathbb{Q}^n$ and semidefinite cone $\mathbb{S}^n_+$ as a subset of $\mathbb{R}^{n(n+1)/2}$, copositive cone $\mathbb{C}^n$ or completely positive cone $\mathbb{P}^n$ (see Example A.10 in the Appendix). For proper cones it holds that their Cartesian product is again proper cone [9, 10].

1.6 Dual problems

1.6.1 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

$$\begin{array}{ll} \text{minimize} & c^T x, \\ \text{subject to} & Ax = b, \\ & x \in \mathcal{K}. \end{array} \tag{1.32}$$

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

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

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 (1.32). Indeed, from the very definition of dual cone:

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

Therefore, the Lagrange dual function is

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

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

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

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 1.

Remark 1.33. For a closed cone $\mathcal{K}$ it holds that $\mathcal{K}^{**} = \mathcal{K}$, where $\mathcal{K}^{**}$ denotes dual cone of the dual cone (see the Proposition A.8 in the Appendix). Therefore, repeating the above procedure, one can easily show that dual of this dual problem is the original primal program (1.32).

1.6.2 Dual problem of QCQP

We will derive dual form of standard QCQP

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

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

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

where

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

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

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

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 $P_0, P_1 \ldots, P_m, q_0, q_1 \ldots, q_m, r_0, r_1 \ldots, r_m$ 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.

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

Due to the Schur complement lemma (see appendix, Theorem A.2) the above is equivalent to

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

where the matrix M is easily expanded as

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

with

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

1.6.3 Second dual of QCQP

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

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

where

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

The Lagrange dual function is

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

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

$$\begin{array}{rcl} Y \bullet M_0 & = & P_0 \bullet X + q_0^T x + r_0, \\ Y \bullet M_k \leq 0 & \Leftrightarrow & P_k \bullet X + q_k^T x + r_k \leq 0, \\ Y \succeq 0 & \Leftrightarrow & X \succeq xx^T. \end{array}$$

Thus we obtain the following SDP as second dual of QCQP

$$\begin{array}{ll} \text{minimize} & P_0 \bullet X + q_0^T x + r_0 \\ \text{subject to} & P_k \bullet X + q_k^T x + r_k \leq 0, \; (k = 1, \dots, m) \\ & X \succeq xx^T. \end{array} \tag{1.34}$$

Chapter 2

Relaxations

It was mentioned in the beginning that our strategy is to relax the nonconvex QCQPs (2) with computationally tractable 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 2.1 (Proposed definition)**.** Let P, Q be a minimization problems

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

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 $g(u(x)) = f(x)$ for all $x \in X$.

Remark 2.2. 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 $g(y') = f(v(y'))$ 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 $g(y') = f(v(y'))$ 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 an identity $u(x) = x, \ \forall x \in X$ (or $Y' = X$ and $v(y') = y', \ \forall y' \in Y'$ for the surjective version) suits the definition.

The plus of this definition is that problems P and Q may have different variable spaces. This is quite common, for example in semidefinite programming relaxation the original QCQP (2) with variable $x \in \mathbb{R}^n$ is relaxed as SDP (2.8) with variable $Y \in \mathbb{S}^{n+1}$.

An interesting point of view is that the relaxation approximates a minimization 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.

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).

2.1 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 has 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 (2). Using identity $x^T P_k x = P_k \bullet xx^T$, which follows from the Definiton 1.14, it can be rewritten as follows

$$\begin{array}{ll} \text{minimize} & P_0 \bullet X + q_0^T x + r_0, \\ \text{subject to} & P_k \bullet X + q_k^T x + r_k \leq 0, \ (k = 1, \ldots, m) \\ & X = xx^T. \end{array} \tag{2.3}$$

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

$$\begin{array}{ll} \text{minimize} & M_0 \bullet Y, \\ \text{subject to} & M_k \bullet Y \leq 0, \ (k = 1, \ldots, m), \\ & X = xx^T, \end{array} \tag{2.4}$$

where ,

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

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 2.5. *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.2 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 2.6. Notice that we have proven the 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 redundant constraints (also known as valid inequalities) is often usefull for strengthening the relaxation. For more on valid inequalities see [4, 17, 18].

Using the Lemma 2.5 we obtain another equivalent formulation of the original QCQP

$$\begin{array}{ll} \text{minimize} & M_0 \bullet Y, \\ \text{subject to} & M_k \bullet Y \le 0,\ (k = 1, \dots, m) \\ & Y \succeq 0, \quad \operatorname{rank} Y = 1. \end{array} \tag{2.7}$$

relaxing the nonconvex rank 1 constrain we get the following SDP relaxation of (2)

$$\begin{array}{ll} \text{minimize} & M_0 \bullet Y, \\ \text{subject to} & M_k \bullet Y \le 0,\ (k = 1, \dots, m) \\ & Y \succeq 0. \end{array} \tag{2.8}$$

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

$$\begin{array}{ll} \text{minimize} & P_0 \bullet X + q_0^T x + r_0, \\ \text{subject to} & P_k \bullet X + q_k^T x + r_k \le 0,\ (k = 1, \dots, m) \\ & X \succeq xx^T. \end{array} \tag{2.9}$$

Remark 2.10. Notice that this is exactly the second dual of QCQP (1.34).

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

Remark 2.11. This is indeed a relaxation (according to our Definition 2.1). Existence of the injective mapping u is obvious from the construction of Y and equivalent problem reformulation (2.4) as

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

2.2 CQP and SOCP relaxation

These relaxations are expected to provide weaker bounds in less time than SDP relaxations. In fact, SOCP relaxations are often constructed as further relaxations of SDP relaxations. In a sense, one can therefore see that SOCP relaxations are never tighter than their SDP counterparts. [5]

Using the procedure from section 1.2, any CQP (an convex instance of QCQP) can be formulated as SOCP. Therefore we can consider any convex quadratic relaxation as SOCP relaxation. In fact, a CQP problems are usually solved using conic primal-dual algorithms for SOCP.

A possible approach to forming such a SOCP (or convex QP) relaxation is by further relaxing the SDP relaxation (2.9).

The reason, why this is considered is that solving SDP is a computationally costly operation and it may not be the best option, even if it offers better bounds. 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 4.2 about branch and bound).

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

2.2.1 SOCP relaxation of semidefinite constraint

Semidefinite constraint $X - xx^T \succeq 0$ in (2.9) is equivalent to $C \bullet (X - xx^T) \geq 0$ for all $C \in S^n_+$. Using this fact, authors of [7] propose SOCP relaxation of the semidefinite constraint $X - xx^T \succeq 0$ by replacing it with multiple constraints of the form

$$x^T C_i x - C_i \bullet X \leq 0 \quad (i = 1, \ldots, t). \tag{2.12}$$

Since $C_i \succeq 0$, these are convex quadratic constraints and using the procedure described earlier (in the Section 1.2.1) one can formulate them equivalently as a second order cone constraints of the form

$$\begin{pmatrix} v_0^i \\ v^i \end{pmatrix} = \begin{pmatrix} 1 + C_i \bullet X \\ 1 - C_i \bullet X \\ 2L_i^T x \end{pmatrix}, \quad ||v^i|| \leq v_0^i, \quad (i = 1, \ldots, l), \tag{2.13}$$

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

$$\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_l \geq 0 > \lambda_{l+1} \geq \cdots \geq \lambda_n, \tag{2.14}$$

be the eigenvalues of the matrix P and let $u_1, \ldots, u_n$ be the corresponding orthonormal eigenvectors. Then spectral decomposition $P = \sum_{j=1}^{h} \lambda_j u_j u_j^T$ implies that $C = C^+ - C^-$, where matrices

$$\begin{aligned} C^+ &= \sum_{j \in J} \lambda_j u_j u_j^T \quad \text{for } J = \{1, \ldots, l\}, \\ C^- &= -\sum_{j \in J} \lambda_j u_j u_j^T \quad \text{for } J = \{l+1, \ldots, n\}, \end{aligned}$$

are positive semidefinite. The matrices C^+ and C^- can be used in (2.13) rewriting $x^T C x - C \bullet X \leq 0$ constraint. We will use this idea later.

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 [7] suggest putting $C_i = e_i e_i^T$, which corresponds to the constraints

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

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

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

2.2.2 SOCP in the original variable

The SOCP relaxation introduced above avoids 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 [7] authors also introduce a general relaxation scheme to obtain SOCP relaxation of nonconvex QCQP (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

$$x^T P_k^+ x + q_k^T x + r_k \leq x^T P_k^- x.$$

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

$$\begin{aligned} x^T P_k^+ x + q_k^T x + r_k &\leq z_k \\ x^T P_k^- x &\leq z_k. \end{aligned}$$

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 [7] described in the Section 1.2.

2.2.3 Relaxing the semidefinite constraint and reducing the number of variables

The idea of combining both above approaches (introduced in [7]) 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

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

Let the eigenvalues λ_i and eigenvectors u_i of P for $(i = 1, \dots, n)$ be as above, see (2.14). And let $P = P^+ - P^-$ with

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

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

$$\begin{array}{l} x^T P^+ x + \sum_{j=l+1}^n \lambda_j z_j + q^T x + r \leq 0, \\ x^T (u_j u_j^T) x - z_j = 0, \ (j = l+1, \dots, n., \end{array} \tag{2.16}$$

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

$$\begin{array}{l} x^T P^+ x + \sum_{j=l+1}^n \lambda_j z_j + q^T x + r \leq 0 \\ x^T (u_j u_j^T) x - z_j \leq 0, \ (j = l+1, \dots, n) \end{array} \tag{2.17}$$

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 (2.16). In fact, if x and z_j for $(j = l+1, \dots, n)$ satisfy (2.16) then

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

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

$$\begin{array}{l} x^T P^+ x + \sum_{j=l+1}^{n} \lambda_j z_j + q^T x + r \leq 0 \\ x^T (u_j u_j^T) x - z_j \leq 0, \ (j = l+1, \ldots, n) \\ \sum_{j=l+1}^{n} z_j \leq \|x\|_2^2 \end{array} \tag{2.18}$$

In the end authors of [7] 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 (2.15) together with valid inequalities relaxing the semidefinite constraint

$$\begin{array}{l} P \bullet X + q^T x + r \leq 0, \\ x^T P^+ x - P^+ \bullet X \leq 0, \\ x^T u_j u_j^T x - u_j u_j^T \bullet X \leq 0, \ (j = l+1, \ldots, n). \end{array} \tag{2.19}$$

Suppose $(x, X) \in \mathbb{R}^n \times \mathbb{S}^n$ satisfies this relaxation, then x and $z_j = u_j^T X u_j$ for $j = l+1, \ldots, n$ satisfy (2.18). So we may see (2.18) as further relaxation of (2.19). 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 [8, 7].

2.3 Mixed SOCP-SDP relaxation

In [5] authors Burer, Kim and Kojima have introduceda SOCP-SDP compromise, relaxation of the QCQP (2) somewhere between SDP and SOCP. We will describe their approach with the special case they provide as introduction. We are dealing with QCQP (2)

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

Let $\lambda_{min}(P_k)$ denote smallest eigenvalue of P_k. For all $k = 0, \ldots, m$ define $\lambda_k = -\lambda_{min}(P_k)$ so that $P_k + I\lambda_k \succeq 0$. Then (2) is equivalent to

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

which has following SOCP-SDP relaxation

$$\begin{array}{ll} \text{minimize} & -\lambda_0 Tr(X) + x^T (P_0 + \lambda_0 I) x + q_0^T x + r_0 \\ \text{subject to} & -\lambda_k Tr(X) + x^T (P_k + \lambda_k I) x + q_k^T x + r_k \leq 0, \\ & (k = 1, \ldots, m) \\ & X \succeq x x^T \end{array} \tag{2.20}$$

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. Moreover, 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 [5] they suppose that $x_j \in [0,1]$ and use $X_{jj} \leq x_j$ to establish boundedness.

Remark 2.21. 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

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

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.

The following proposition from [44] gives an equivalent formulation of $X \succeq xx^T$ constraint, only in terms of x and diagonal entries of X.

Proposition 2.22 ([44])**.** *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 (2.20) is equivalent to

$$\begin{array}{ll} \text{minimize} & -\lambda_0 Tr(X) + x^T(P_0 + \lambda_0 I)x + q_0^T x + r_0 \\ \text{subject to} & -\lambda_k Tr(X) + x^T(P_k + \lambda_k I)x + q_k^T x + r_k \leq 0, \\ & (k = 1, \dots, m) \\ & x_j^2 \leq X_{jj} \leq x_j \quad (j = 1, \dots, n) \end{array} \tag{2.23}$$

Compared to SDP relaxation (2.9), which has $O(n^2)$ variables, problem (2.23) has only $O(n)$ and hence is much faster to solve. On the other hand bound should be generally weaker than the SDP bound.

Block diagonal splitting

This approach is further generalized and explored in [5]. They consider more general splitting $P = -D + (P + D)$, instead of $-\lambda I + (P + \lambda I)$, where D is some permutation of a block diagonal matrix.

Let $r \leq n$ be a positive integer and $\mathcal{C} = \{C_1, \dots, C_r\}$ be a partition of the indices $\{1, \dots, n\}$. We will say matrix D is C-block diagonal matrix if $D_{ij} = 0$ whenever i and j are members of different sets in $\mathcal{C}$.

Based on partition $\mathcal{C}$ authors construct a mixed SOCP-SDP relaxation as follows. For each $k = 0, \dots, m$ let D_k be a $\mathcal{C}$-block diagonal matrix satisfying $P_k + D_k \succeq 0$.

this yields following problem equivalent to (2)

$$\begin{array}{ll} \text{minimize} & -x^T D_0 x + x^T(P_0 + D_0)x + q_0^T x + r_0 \\ \text{subject to} & -x^T D_k x + x^T(P_k + D_k)x + q_k^T x + r_k \le 0, \ (k = 1, \dots, m), \end{array}$$

and its mixed SOCP-SDP relaxation is

$$\begin{array}{ll} \text{minimize} & -D_0 \bullet X + x^T(P_0 + D_0)x + q_0^T x + r_0 \\ \text{subject to} & -D_k \bullet X + x^T(P_k + D_k)x + q_k^T x + r_k \le 0, \ (k = 1, \dots, m), \\ & X \succeq xx^T. \end{array}$$

The following proposition generalizes Proposition 2.22.

Proposition 2.24 ([44]). *Given a vector x and symmetric blocs $X_{C_1C_1}, \dots, X_{C_rC_r}$, there exists a symmetric-matrix completion $X \in S^n$ of $X_{C_1C_1}, \dots, X_{C_rC_r}$, satisfying $X \succeq xx^T$ if and only if $X_{C_jC_j} \succeq x_{C_j}x_{C_j}^T$ for all $j = 1, \dots r$.*

Therefore we can simplify our relaxation to

$$\begin{array}{ll} \text{minimize} & -\sum_{j=1}^r [D_0]_{C_jC_j} \bullet X_{C_jC_j} + x^T(P_0 + D_0)x + q_0^T x + r_0 \\ \text{subject to} & -\sum_{j=1}^r [D_k]_{C_jC_j} \bullet X_{C_jC_j} + x^T(P_k + D_k)x + q_k^T x + r_k \le 0, \\ & \qquad (k = 1, \dots, m), \\ & X_{C_jC_j} \succeq x_{C_j}x_{C_j}^T \ (j = 1, \dots, r). \end{array} \tag{2.25}$$

Again, considered entries of X should be bounded from above, for the relaxation to make sense.

Authors also propose method to choose such $\mathcal{C}$-block diagonal matrices D_k. For matrix M and partition of indices $\mathcal{C}$ denote $M(\mathcal{C})$ a $\mathcal{C}$-block diagonal matrix with $M_{ij} = M(\mathcal{C})_{ij}$ for all i, j contained in the same partition of $\mathcal{C}$. And denote $\bar{M}(\mathcal{C}) = M - M(\mathcal{C})$ the matrix with all the complementary entries of M.

Their initial choice is either

$$D_k = -\lambda_{min}(P_k)I \quad \text{or} \quad D_k = -\lambda_{min}(\bar{P}_k(\mathcal{C}))I - P_k(\mathcal{C}).$$

It is clear that $(P_k + D_k) \succeq 0$ for each of these choices, for first one it is trivial and for second one we have

$$P_k + D_k = -\lambda_{min}(\bar{P}_k(\mathcal{C}))I - \bar{P}_k(\mathcal{C}) \succeq 0.$$

Authors in [5] introduce an algorithm which improves this choice of D_k.

2.4 LP relaxation of QCQP

Semidefinite program relaxations can provide tight bounds, but these 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 (2.9) replacing the semidefinite constraint $X \succeq xx^T$ with symmetry constraint $X = X^T$.

$$\begin{array}{ll} \text{minimize} & P_0 \bullet X + q_0^T x + r_0 \\ \text{subject to} & P_k \bullet X + q_k^T x + r_k \le 0, \ (k = 1, \dots, m) \\ & X = X^T. \end{array} \tag{2.26}$$

Suppose that the feasible set in (2.26) 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 $0 \le x_i \le 1$ box constraint). We will separate all such box constraints in the next formulation.

$$\begin{array}{ll} \text{minimize} & P_0 \bullet X + q_0^T x + r_0 \\ \text{subject to} & P_k \bullet X + q_k^T x + r_k \le 0, \ (k = 1, \dots, m) \\ & l_i \le x_i \le u_i, \ (i = 1, \dots, n) \\ & X = X^T. \end{array} \tag{2.27}$$

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

2.4.1 Reformulation linearization technique (RLT)

The optimal value of (2.27) 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 [13, 12, 14]. It adds linear inequalities to (2.27). 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 (2.27). By taking into account only the four original bounds $x_i - l_i \ge 0$, $x_i - u_i \le 0$, $x_j - l_j \ge 0$, $x_j - u_j \le 0$, we get the RLT inequalities

$$\begin{array}{rcl} X_{ij} - u_i x_j - u_j x_i & \ge & -u_i u_j, \\ X_{ij} - u_i x_j - l_j x_i & \le & -u_i l_j, \\ X_{ij} - l_i x_j - u_j x_i & \le & -l_i u_j, \\ X_{ij} - l_i x_j - l_j x_i & \ge & -l_i l_j. \end{array} \tag{2.28}$$

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

$$\begin{array}{ll} \text{minimize} & P_0 \bullet X + q_0^T x + r_0 \\ \text{subject to} & P_k \bullet X + q_k^T x + r_k \leq 0, \ (k = 1, \dots, m) \\ & X - lx^T - xl^T \quad \geq \quad -ll^T, \\ & X - ux^T - xu^T \quad \geq \quad -uu^T, \\ & X - lx^T - xu^T \quad \leq \quad -lu^T, \\ & l \leq x \leq u, \quad X = X^T, \end{array} \tag{2.29}$$

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 can be imposed on X.

2.4.2 Positive semidefinite cuts

In the above LP relaxations we have relaxed nonconvex $X = xx^T$ constraint in (2.3) 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 (2.8), i.e.

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

where

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

From the definition of positive semidefinite matrix it holds that

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

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.

$$\begin{array}{ll} \text{minimize} & M_0 \bullet Y \\ \text{subject to} & M_k \bullet Y \leq 0, \ (k = 1, \dots, m) \\ & v^T Y v \geq 0, \ v \in V, \end{array} \tag{2.30}$$

for some finite set of vectors V.

The usual approach is declaring such PSD cuts with respect to a feasible solution $\bar{Y}$ of the relaxation (2.30) (for example $\bar{Y}$ may be the optimal solution of (2.30)). 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 (2.30) 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 [12] 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 $\bar{v}^T \bar{Y} \bar{v} \leq 0$.

2.5 Strengthening the SDP relaxation

In the previous section we have 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 additional valid inequalities (relaxation of semidefinite constraint, RLT or PSD cuts). Here, we offer an example of a redundant inequality tightening the relaxation.

Example 2.31. Consider following problem

$$\begin{array}{ll} \text{maximize} & x^T L x, \\ \text{subject to} & x \in \{-1,1\}^n, \end{array} \tag{2.32}$$

for

$$n = 4 \quad \text{and} \quad L = \begin{pmatrix} 1 & 0 & 0 & -1 \\ 0 & 3 & -1 & -2 \\ 0 & -1 & 2 & -1 \\ -1 & -2 & -1 & 4 \end{pmatrix}.$$

One can easily form an SDP relaxation

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

and find out the optimum using any solver for SDP

$$L \bullet \hat{X}_1 = 16.5 \quad \text{for} \quad \hat{X}_1 = \begin{pmatrix} 1 & 0.875 & 0.25 & -1 \\ 0.875 & 1 & -0.25 & -0.875 \\ 0.25 & -0.25 & 1 & -0.25 \\ -1 & -0.875 & -0.25 & 1 \end{pmatrix}.$$

On the other hand, when the redundant constraint

$$x_2x_3 + x_3x_4 + x_4x_2 \geq -1,$$

is added to the original problem (2.32), an then relaxed together with the problem as

$$\begin{array}{ll} \text{maximize} & L \bullet X, \\ \text{subject to} & diag(X) = 1, \quad X \succeq 0, \\ & X_{23} + X_{34} + X_{42} \geq -1, \end{array}$$

the obtained optimum is better, in fact it is

$$L \bullet \hat{X}_2 = 16 \quad \text{for} \quad \hat{X}_2 = \begin{pmatrix} 1 & 1 & -0.0524 & -1 \\ 1 & 1 & -0.0523 & -1 \\ -0.0524 & -0.0523 & 1 & 0.0523 \\ -1 & -1 & 0.0523 & 1 \end{pmatrix},$$

which is equal to the optimal value of the original problem for $\hat{x} = (1, 1, 1, -1)^T$. We will come back to this example later and explain where does the magic constraint come from (see the Section 3.3.2).

2.5.1 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 may be surprising that RLT inequalities also help when they are added to SDP relaxation. In fact Anstreicher in [13] showed that following RLT + SDP relaxation is stronger than both SDP or RLT relaxations.

$$\begin{array}{ll} \text{minimize} & P_0 \bullet X + q_0^T x + r_0 \\ \text{subject to} & P_k \bullet X + q_k^T x + r_k \leq 0, \ (k = 1, \dots, m) \\ & X - lx^T - xl^T \quad \geq \quad -ll^T, \\ & X - ux^T - xu^T \quad \geq \quad -uu^T, \\ & X - lx^T - xu^T \quad \leq \quad -lu^T, \\ & l \leq x \leq u, \quad X \succeq xx^T. \end{array} \tag{2.33}$$

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 [15] 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 [7] offers a way to avoid the cost of combining SDP and LP constraints.

2.6 Copositive and completely positive programming representation

Consider a special case of QCQP 2 a class of linearly constrained quadratic optimization problems in continuous nonnegative and binary variables

$$\begin{array}{ll} \text{minimize} & x^T Q x + 2c^T x, \\ \text{subject to} & Ax = b, \\ & x_j(1 - x_j) = 0 \ (j \in B), \\ & x \in \mathbb{R}^n_+. \end{array} \tag{2.34}$$

In 2009, completely positive programming (CPP) relaxation for this class of QOPs was proposed by Burer [20]

$$\begin{array}{ll} \text{minimize} & Q \bullet X + 2c^T x, \\ \text{subject to} & Ax = b, \\ & a_i X a_i = b_i^2 \ (i = 1, \dots, n), \\ & x_j = X_{jj} \ (j \in B), \\ & \begin{pmatrix} 1 & x^T \\ x & X \end{pmatrix} \in \mathbb{P}^{n+1}, \end{array} \tag{2.35}$$

where $X \in \mathbb{S}^n$, $x \in \mathbb{R}^n$ are variables, $\mathbb{P}^{n+1}$ is a completely positive cone (see Definition 1.20) and Q, A, c, b, B are from original problem.

As shown in [20] the optimal value of this relaxation coincides with optimal value of (2.34).

The class was extended to a more general class of QOPs by Eichfelder and Povh [21] and by Arima, Kim and Kojima [19]. Theoretically strong results were presented in their papers [19, 20, 21] 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 (2.34) involves various NP-hard combinatorial problems (such as max-cut), therefore solving these CPP programs is also NP-hard.

2.6.1 The Lagrangian doubly nonnegative relaxation

In this section we will describe relaxation by Kim, Kojima and Toh [22] of the more general problem with complementarity constraints

$$\begin{array}{ll} \text{minimize} & u^T Q u + 2c^T u, \\ \text{subject to} & Au = b, \\ & u_i u_j = 0, \ ((i, j) \in \mathcal{E}), \\ & u \in \mathbb{R}^n_+, \end{array} \tag{2.36}$$

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 $v_i = (1-u_i) \geq 0$. In [22] authors assume that linear constraint set $\{u \in \mathbb{R}^n_+ \mid Au = b\}$ is bounded.

The problem (2.36) is first reformulated as

$$\begin{array}{ll} \text{minimize} & Q_0 \bullet xx^T, \\ \text{subject to} & H_0 \bullet xx^T = 1, \\ & Q_{01} \bullet xx^T = 0, \\ & Q_{ij} \bullet xx^T = 0,\ ((i,j) \in \mathcal{E}), \\ & x \in \mathbb{R}^{n+1}_+, \end{array} \tag{2.37}$$

where

$$\begin{array}{rcl} x & = & (x_1, u) \in \mathbb{R}^{n+1}_+ \\ Q_0 & = & \begin{pmatrix} 0 & c^T \\ c & Q \end{pmatrix} \in \mathbb{S}^{n+1}, \\ H_0 & = & \begin{pmatrix} 1 & 0^T \\ 0 & O \end{pmatrix} \in \mathbb{S}^{n+1}, \\ Q_{01} & = & \begin{pmatrix} b^T b & b^T A \\ A^T b & A^T A \end{pmatrix} \in \mathbb{S}^{n+1}, \\ Q_{ij} & = & \begin{pmatrix} 0 & 0^T \\ 0 & C_{ij} + Cij^T \end{pmatrix} \in \mathbb{S}^{n+1}, \\ C_{ij} & = & \begin{array}{l} \text{the } n \times n \text{ matrix with } (i,j)\text{-th element } 1/2 \\ \text{and } 0 \text{ elsewhere.} \end{array} \end{array}$$

Since the matrices Q_{ij} for $(i,j) \in \{(0,1)\} \cup \mathcal{E}$ are nonegative, $Q_{ij} \bullet xx^T \geq 0$ holds for any $x \in \mathbb{R}^n_+$. Hence, the set of equalities $Q_{ij} \bullet xx^T = 0$ can be combined into a single equality $H_1 \bullet xx^T = 0$, where $H_1 = \sum Q_{ij}$ through $(i,j) \in \{(0,1)\} \cup \mathcal{E}$. Consequently they obtain simplified problem

$$\begin{array}{ll} \text{minimize} & Q_0 \bullet xx^T, \\ \text{subject to} & H_0 \bullet xx^T = 1, \\ & H_1 \bullet xx^T = 0, \\ & x \in \mathbb{R}^n_+, \end{array} \tag{2.38}$$

which is equivalent to (2.36). 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 [22] relaxed as

$$\begin{array}{ll} \text{inf} & Q_0 \bullet X, \\ \text{subject to} & H_0 \bullet X = 1, \\ & H_1 \bullet X = 0, \\ & X \in \mathcal{K}, \end{array} \tag{2.39}$$

where $\Gamma \subseteq \mathcal{K}$. The cone $\mathcal{K}$ can be chosen as any of the Γ, $\mathbb{P}^n$, $\mathbb{S}^n \cap \mathbb{N}^n$, $\mathbb{S}^n$, $\mathbb{S}^n + \mathbb{N}^n$, $\mathbb{C}^n$ (see the Proposition 1.25). 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 2.36 is that its Lagrangian relaxation

$$\begin{aligned} L^p(\lambda,\mathcal{K}) &:= \inf\{Q_0 \bullet X + \lambda H_1 \bullet X \mid H_0 \bullet X = 1,\ X \in \mathcal{K}\} \qquad (2.40)\\ L^d(\lambda,\mathcal{K}) &:= \sup\{y_0 \mid Q_0 + \lambda H_1 - y_0 H_0 \in \mathcal{K}^*\} \qquad (2.41)\end{aligned}$$

has dual with only one variable. As they show in [22] 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 (2.36).

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 (2.36).

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 [22] authors Kim, Kojima and Toh proposed a fast algorithm for solving Lagrangian DNN relaxation (2.40 - 2.41) based on combination of bisection method and first order methods. Recently (February 2016), Arima, Kim, Kojima and Toh [23] 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.

2.7 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 valid inequalities that may be helpful for strengthening an LP or SDP, there is a more systematic (and potentially more powerful) approach consisting in the use of LP or SDP hierarchies. In particular there are procedures by Balas, Ceria, Cornuéjols [26]; Lovász, Schrijver [27] (with LP-strengthening LS and an SDP-strengthening LS+); Sherali, Adams [28] or Lasserre [29, 30].

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 [31].

Consider the following binary polynomial optimization problem

$$\begin{array}{ll} \text{minimize} & p_0(x),\\ \text{subject to} & p_k(x) \le 0,\ (k = 1, \dots, m)\\ & x \in \{0,1\}^n. \end{array} \qquad (2.42)$$

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

$$1, x_1, \ \ldots, \ x_n, \ x_1^2, \ x_1x_2, \ \ldots, \ x_n^2, \ \ldots, \ x_1^r, \ \ldots, \ x_n^r, \tag{2.43}$$

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 (2.43). Hence the vectors of coefficients of the polynomials $p_k(x)$ in (2.42) are p_k for $k = 1, \ldots, m$.

Denote $V = \{1, \ldots, n\}$, $\mathcal{P}(V)$ the collection of all subsets of V and $\mathcal{P}_t(V)$ for $1 \leq t \leq n$ the collection of all subsets of cardinality $\leq t$. 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

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

Definition 2.44 (Moment and Localizing matrix)**.** Given a vector $y \in \mathbb{R}^{\mathcal{P}(V)}$ and an integer t, $1 \leq t \leq n$ the matrices

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

are known as moment matrix and localizing matrices respectively.

The Lasserre relaxations are based on following observation

Lemma 2.45. *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*

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

Proof. Indeed, for any $I, J \subseteq V$ with $|I|, |J| \leq 2t$, 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 2.46 (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

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

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

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

The relaxed problem then takes form

$$\begin{array}{ll} \text{minimize} & \sum_\alpha (p_0)_\alpha y_\alpha \\ \text{subject to} & y \in L_t(K). \end{array} \tag{2.47}$$

Let us denote p_t^* the optimal value of t-th round Lasserre relaxation. In [29] Lasserre shows that the sequence p_t^* converges to optimal value of the original problem (2.42). Furthermore, the convergence is finite.

In fact, Lasserre states that every program (2.42) is equivalent to SDP with $2^n - 1$ (because any $L_t(K)$ can be reduced to $2^n - 1$ variables). And the projection of the feasible set $L_n(K)$ 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. The hierarchy, however has a potential to offer valuable insights. In the survey [32] 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] (we will cover their analysis later in the Section 3.2).

On the other hand, the hierarchy of Lasserre might be a good measure of hardness of problem instances. In [33] 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 3

The maximum cut problem

The aim of this chapter is 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 nonconvex QCQP. Then, we will describe in detail the famous analysis by Goemans and Williamson [37] showing that 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 2 can be applied to max-cut problem. And will provide a computational comparison of these approaches.

We have chosen this particular problem for two reasons. The first reason is 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 a benchmark for evaluating the performance of other relaxation methods, most of the approaches mentioned in the Chapter 2 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 also be an insightful observation.

3.1 Problem formulation

Problem statement (max-cut). Let $G = (V, \mathcal{E})$ be an undirected graph where $V = \{1, \dots, n\}$ 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.

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

we define

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

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

3.1.1 Quadratic 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

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

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

$$\begin{array}{ll} \text{maximize} & \frac{1}{2}\sum_{i<j} w_{ij}(1 - x_i x_j), \\ \text{subject to} & x \in \{-1, 1\}^n. \end{array} \tag{3.1}$$

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

$$L = diag(We) - W, \tag{3.2}$$

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

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

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

$$\begin{array}{ll} \text{maximize} & \frac{1}{4}x^T L x, \\ \text{subject to} & x^T e_i e_i^T x = 1, \ (i = 1, \ldots, n). \end{array} \tag{3.3}$$

Therefore max-cut is indeed an instance of QCQP.

3.2 Goemans and Williamson analysis

In their famous article [37] authors first relax the problem (3.1) 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

$$\begin{array}{ll} \text{maximize} & \frac{1}{2}\sum_{i<j} w_{ij}(1 - v_i^T v_j), \\ \text{subject to} & v_i \in S^n \ (i = 1, \dots, n). \end{array} \tag{3.4}$$

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 (3.4), 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 the step 2, a 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 3.5 ([37]). *Let ω be the value of the cut produced by the above GW-algorithm and $E[\omega]$ its expected value. Authors show that*

$$E[\omega] = \frac{1}{\pi}\sum_{i<j} w_{ij} \arccos(v_i^T v_j), \tag{3.6}$$

and

$$E[\omega] \geq \alpha\frac{1}{2}\sum_{i<j} w_{ij}(1 - v_i^T v_j) \quad \textit{for} \quad \alpha = 0.87856. \tag{3.7}$$

Proof. First we will show (3.6). By definition and linearity of expected value in the first step, using symmetry in the second we have

$$\begin{align} E[\omega] &= \sum_{i<j} w_{ij} \ Pr\left[sgn(v_i^T r) \neq sgn(v_j^T r)\right] \tag{3.8} \\ &= \sum_{i<j} w_{ij} \ 2Pr\left[v_i^T r \geq 0 \ \wedge \ v_j^T r < 0\right]. \tag{3.9} \end{align}$$

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,

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

Plugging this identity into (3.9), the first part follows.

Secondly we will show (3.7). It is enough to prove that

$$\frac{1}{\pi}\arccos(v_i^T v_j) \geq \alpha\frac{1}{2}(1 - v_i^T v_j), \tag{3.10}$$

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

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

where the left hand side is equal to $E[\omega]$ from (3.6).

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 (3.10) reduces to

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

Which is just a simple exercise to prove.

□

Remark 3.11. Denote the objective values

ω_r	–	value in the optimal solution of the relaxation (3.4)
ω^*	–	value in the optimal solution of the maxcut problem (3.1)
ω	–	value in the feasible solution of (3.1) obtained by the GW-algorithm

Then the following inequality holds trivially,

$$\omega \leq \omega^* \leq \omega_{.}$$

The proposition 3.5 further implies

$$0.878\ \omega_r < E[\omega] \leq \omega^* \leq \omega_r.$$

Therefore, the bounds obtained by solving the relaxed problem are quite tight.Furthermore, the feasible solutions projected by GW-algorithm are also good in expectation. Generally, we need to solve the relaxation (3.4) only once and then repeat steps 2 and 3 of the algorithm couple of times. Keeping only the best solution will (probably) get the objective at least as good as the expected value.

3.2.1 Semidefinite relaxation of the max-cut

Now the last arising question is how do we solve the relaxed problem (3.4)? Let us remind the formulation of the relaxed problem

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

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 $\{1, \dots, n\}$.

Let $B := (v_1, \dots, v_n)$ 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 $i, j \in (1, \dots, n)$, 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 $X = BB^T$.

$$\begin{array}{ll} \text{maximize} & \frac{1}{4}L \bullet X, \\ \text{subject to} & diag(X) = e \\ & X \succeq 0, \end{array} \tag{3.12}$$

where $X \in \mathbb{S}_+^n$ is a variable, the matrix L is defined in (3.2), $diag(X)$ is the vector of diagonal entries of X, and e is the n-dimensional vector of ones.

Remark 3.13. In fact, we would receive the same semidefinite relaxation by following the general scheme for QCQP described in Section 2.1.

3.3 Relaxations of the max-cut

3.3.1 Semidefinite relaxation

W will derive the SDP relaxation for max-cut once again, now using the procedure for QCQP described in Section 2.1.

First introduce a new variable $X = xx^T$ in (3.1) and rewrite this constraint as $X \succeq 0$, and rank $X = 1$. The maximum cut problem is then equivalent to

$$\begin{array}{ll} \text{maximize} & \frac{1}{4}L \bullet X, \\ \text{subject to} & e_i e_i^T \bullet X = 1, \ (i = 1, \dots, n), \\ & X \succeq 0, \quad \text{rank } X = 1. \end{array} \tag{3.14}$$

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. Relaxing the nonconvex rank 1 constraint we obtain (3.12).

Adding RLT constraints

Since max-cut is symmetric in a sense that the objective value for x and $-x$ are equal, there is no loss of generality if we fix $x_1 = 1$, i.e. the first vertex will belong to S.

Let X_1 denote the first column of X. Putting $x_1 = 1$ into the original constraint $X = xx^T$ yields $X_1 = x$. Therefore, even if we do not use the variable x explicitly, we can still form RLT constraints. These constraints will create links between first column (first row due to symmetry) and the rest of the matrix X.

Since the original variable x is in $\{-1, 1\}^n$, the redundant box constraints $-e \leq x \leq e$ can be added. Multiplying pairs of these constraints and relaxing them (see the Section 2.4.1) we obtain following valid RLT (element-wise) inequalities

$$\begin{array}{rcl} X + X_1e^T + eX_1^T & \geq & -ee^T, \\ X - X_1e^T - eX_1^T & \geq & -ee^T, \\ X - X_1e^T + eX_1^T & \geq & -ee^T. \end{array} \tag{3.15}$$

Adding them to (3.12) we will obtain SDP + RLT relaxation

$$\begin{array}{ll} \text{maximize} & \frac{1}{4}L \bullet X, \\ \text{subject to} & diag(X) = e, \\ & X + X_1e^T + eX_1^T \;\geq\; -ee^T, \\ & X - X_1e^T - eX_1^T \;\geq\; -ee^T, \\ & X - X_1e^T + eX_1^T \;\geq\; -ee^T, \\ & X \succeq 0. \end{array} \tag{3.16}$$

3.3.2 Triangle inequalities

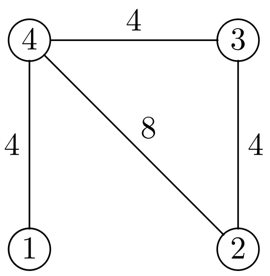
In the above reformulation of max-cut (3.14), it is clear that following inequalities hold for X and any triple of vertices $i, j, k \in V$

$$\begin{array}{rcl} X_{ij} + X_{jk} + X_{ik} & \geq & -1, \\ X_{ij} - X_{jk} - X_{ik} & \geq & -1, \\ -X_{ij} - X_{jk} + X_{ik} & \geq & -1, \\ -X_{ij} + X_{jk} - X_{ik} & \geq & -1 \end{array} \tag{3.17}$$

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, however on the other hand, large number of linear constraints will also increase the computational time.

Example 3.18. Recall the Example 2.31 of strengthening SDP by adding a valid inequality. In fact, that example corresponds to the max-cut instance with L representing a following graph



$$L = \begin{pmatrix} 1 & 0 & 0 & -1 \\ 0 & 3 & -1 & -2 \\ 0 & -1 & 2 & -1 \\ -1 & -2 & -1 & 4 \end{pmatrix}$$

and the added redundant constraint $x_2x_3+x_3x_4+x_4x_2 \geq -1$, represents the triangle inequality for the triangle $(2,3,4)$ contained in the graph.

3.3.3 SOCP relaxations

Let

$$\lambda_1 \leq \cdots \leq \lambda_l \leq 0 < \lambda_{l+1} \leq \cdots \geq \lambda_n,$$

be the eigenvalues of matrix L and let $q_1, \ldots, q_n$ 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 [7] which we have described in the Section 2.2.3 we obtain following SOCP relaxation of max-cut

$$\begin{array}{ll} \text{maximize} & t, \\ \text{subject to} & \frac{1}{4}x^T L^- x + \sum_{j=l+1}^{n} \lambda_j z_j + t \leq 0, \\ & x^T q_j q_j^T x - z_j \leq 0, \ (j = l+1, \ldots, n), \\ & x^T e_j e_j^T x \leq 1, \ (j = 1, \ldots, n), \\ & z_j \leq \sqrt{n}, \ (j = l+1, \ldots, n). \end{array} \tag{3.19}$$

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

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

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 $C \bullet X - x^T C x \geq 0$.

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

$$\begin{aligned} &e_i e_i^T, && i = 1, \ldots, n, \\ &u_{ij} u_{ij}^T, && [i,j] \in \mathcal{E}, \\ &v_{ij} v_{ij}^T, && [i,j] \in \mathcal{E}, \end{aligned}$$

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

$$\begin{aligned} x^T e_i e_i^T x - e_i e_i^T \bullet X &\leq 0,\ i = 1, \ldots, n, && (3.20) \\ x^T u_{ij} e_{ij}^T x - u_{ij} u_{ij}^T \bullet X &\leq 0,\ [i,j] \in \mathcal{E}, && (3.21) \\ x^T v_{ij} e_{ij}^T x - u_{ij} v_{ij}^T \bullet X &\leq 0,\ [i,j] \in \mathcal{E}. && (3.22) \end{aligned}$$

As Kim and Kojima in [7], they also reduce the number of variables. In fact, since $X_{ii} = 1$, the constraint (3.20) reduces to $x_i^2 \leq 1$. By introducing the new variables

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

they obtain a quadratic inequalities from (3.21) and (3.22)

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

For those variables they have following bound

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

Furthermore, they prove following proposition

Proposition 3.23.

$$L = - \sum_{[i,j] \in \mathcal{E}} L_{ij} v_{ij} v_{ij}^T. \tag{3.24}$$

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 (3.24) is

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

□

Using the proposition the objective is rewritten as

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

Now with X removed they obtain the following relaxation

$$\begin{array}{ll}
\text{maximize} & -\frac{1}{4}\sum_{[i,j]\in\mathcal{E}} L_{ij}z_{ij}, \\
\text{subject to} & x_i^2 \le 1,\ (i=1,\dots,n), \\
& (x_i+x_j)^2 \le s_{ij},\ [i,j]\in\mathcal{E}, \\
& (x_i-x_j)^2 \le y_{ij},\ [i,j]\in\mathcal{E}, \\
& s_{ij}+z_{ij}=4,\ [i,j]\in\mathcal{E}.
\end{array} \tag{3.25}$$

They also propose adding a certain form of triangle inequalities into the relaxation (3.25).

Proposition 3.26. *If $[i,j],\ [j,k],\ [k,l]\in\mathcal{E}$, then following inequalities are valid*

$$\begin{aligned}
z_{ij}+z_{jk}+z_{ki} &\le 8, \\
z_{ij}+s_{jk}+s_{ki} &\le 8, \\
s_{ij}+s_{jk}+z_{ki} &\le 8, \\
s_{ij}+z_{jk}+s_{ki} &\le 8.
\end{aligned}$$

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

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

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

Putting all together (triangle inequalities and (3.25) we have one more SOCP relaxation for max-cut

$$\begin{array}{ll} \text{maximize} & -\frac{1}{4}\sum_{[i,j]\in\mathcal{E}} L_{ij}z_{ij}, \\ \text{subject to} & x_i^2 \le 1,\ (i=1,\dots,n), \\ & \left.\begin{array}{l} (x_i+x_j)^2 \le s_{ij}, \\ (x_i-x_j)^2 \le y_{ij}, \\ s_{ij}+z_{ij}=4, \end{array}\right\} [i,j]\in\mathcal{E}, \\ & \left.\begin{array}{lcl} z_{ij}+z_{jk}+z_{ki} & \le & 8, \\ z_{ij}+s_{jk}+s_{ki} & \le & 8, \\ s_{ij}+s_{jk}+z_{ki} & \le & 8, \\ s_{ij}+z_{jk}+s_{ki} & \le & 8. \end{array}\right\} [i,j],\ [j,k],\ [k,l]\in\mathcal{E}. \end{array} \tag{3.27}$$

As the number of the variables in these relaxations (3.25), (3.27) 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.

3.3.4 Mixed SOCP-SDP relaxation

We will form a relaxation based on the approach by Burer, Kim and Kojima [5], which we have described in section 2.3.

In this problem we are maximizing, therefore instead of looking for the smallest eigenvalue of we will need the largest one. Let λ_0 be the largest eigenvalue of L.

By splitting the matrix L we will rewrite the max-cut as follows

$$\begin{array}{ll} \text{maximize} & \frac{1}{4}\left[\ x^T(L-\lambda_0 I_n)x + \lambda_0 x^T I_n x\ \right], \\ \text{subject to} & diag(X)=e. \end{array}$$

And using the procedure from Section 2.3 we obtain a following relaxation

$$\begin{array}{ll} \text{maximize} & \frac{1}{4}\left[\ x^T(L-\lambda_0 I_n)x + \lambda_0 Tr(X)\ \right], \\ \text{subject to} & diag(X)=e, \\ & x_i^2 \le X_{ii}\ (i=1,\dots,n). \end{array}$$

Notice, that using the constraint $diag(X)=e$ we know that $Tr(X)=n$ and $x_i^2 \le X_{ii}$ reduces to $x_i^2 \le 1$. Finally we are left with

$$\begin{array}{ll} \text{maximize} & \frac{1}{4}\left[\ x^T(L-\lambda_0 I_n)x + \lambda_0 n\ \right], \\ \text{subject to} & diag(X)=e, \\ & x_i^2 \le 1\ (i=1,\dots,n). \end{array} \tag{3.28}$$

Block diagonal splitting

Using the more general $\mathcal{C}$-block diagonal splitting of matrix L by [5], we will obtain another relaxation which was described in Section 2.3. Now we will apply this technique to the max-cut.

Let $\mathcal{C} = \{C_1, \dots, C_r\}$ be a partition of the indices $\{1, \dots, n\}$. Let M be a $n \times n$ matrix. We will denote $M(\mathcal{C})$ a $n \times n$ matrix with entries

$$M(\mathcal{C})_{ij} = \begin{cases} M_{ij}, & \text{if } i, j \text{ are in the same partition of } \mathcal{C}, \\ 0, & \text{otherwise.} \end{cases}$$

We denote $\bar{M}(\mathcal{C}) = M - M(\mathcal{C})$ the matrix with all the complementary entries of M. And denote $\lambda_{max}(M)$ the largest eigenvalue of M.

Let

$$D = L(\mathcal{C}) + \lambda_{max}(\bar{L}(\mathcal{C}))I.$$

We will be splitting the matrix L as $L = (L - D) + D$, because we are maximizing, we need $(L - D)$ to be negative semidefinite, which it indeed is

$$(L - D) = \bar{L}(\mathcal{C}) - \lambda_{max}(\bar{L}(\mathcal{C}))I \preceq 0.$$

Hence, we have the following mixed SOCP-SDP relaxation

$$\begin{array}{ll} \text{maximize} & \frac{1}{4}\left[\; x^T(L-D)x + \sum_{j=1}^r D_{C_jC_j} \bullet X_{C_jC_j} \;\right], \\ \text{subject to} & diag(X) = e, \\ & X_{C_jC_j} \succeq x_{C_j}x_{C_j}^T \; (j = 1, \dots, r). \end{array} \tag{3.29}$$

In the special case when $r = n$ we have splitting $L = (\bar{D}_L - \lambda_{max}(\bar{D}_L)I) + (D_L + \lambda_{max}(\bar{D}_L)I)$, where D_L denotes the diagonal matrix with diagonal entries of L and $\bar{D}_L = L - D_L$ denotes its complement. Using that $diag(X) = e$ the resulting relaxation for this splitting is

$$\begin{array}{ll} \text{maximize} & \frac{1}{4}\left[\; x^T(\bar{D}_L - \lambda_{max}(\bar{D}_L)I)x + (e^T D_L + \lambda_{max}(\bar{D}_L)n) \;\right], \\ \text{subject to} & x_i^2 \leq 1, \; (i = 1, \dots, n). \end{array} \tag{3.30}$$

3.3.5 LP relaxations

Following the form introduced in the Section 2.4 we will form linear programming relaxations for max-cut. These relaxations are expected to be fast and produce weak bounds.

Standard LP relaxation

The standard relaxation is easily obtained form the SDP relaxation (3.12) omitting the semidefinite constraint

$$\begin{array}{ll} \text{maximize} & \frac{1}{4}L \bullet X, \\ \text{subject to} & diag(X) = e, \\ & X = X^T. \end{array} \tag{3.31}$$

This relaxation can be strengthened by adding the triangle inequalities (3.17) resulting in

$$\begin{array}{ll} \text{maximize} & \frac{1}{4}L \bullet X, \\ \text{subject to} & diag(X) = e, \\ & X = X^T, \\ & \left.\begin{array}{rcl} X_{ij} + X_{jk} + X_{ik} & \geq & -1, \\ X_{ij} - X_{jk} - X_{ik} & \geq & -1, \\ -X_{ij} - X_{jk} - X_{ik} & \geq & -1, \\ -X_{ij} + X_{jk} - X_{ik} & \geq & -1. \end{array}\right\} i,j,k \in V \end{array} \tag{3.32}$$

LP with RLT relaxation

As we have mentioned earlier in the Section 2.4.1 a reformulation linearization techinque provides means to strengthen the relaxations.

Since max-cut is symmetric in a sense that objective value for x and $-x$ are equal, there is no loss of generality if we fix $x_1 = 1$, i.e. first vertex will belong to S.

Let X_1 denote the first column of X. Putting $x_1 = 1$ into the original constraint $X = xx^T$ yields $X_1 = x$. Therefore, even if we do not use variable x explicitly, we can still form RLT constraints building links between first column (or first row due to symmetry) and the rest of the matrix X.

Since $x \in \mathbb{R}^n$ represents original variable $x \in \{0,1\}^n$ we can use box constraints $-e \leq x \leq e$ and obtain following RLT constraints linking X and X_1 variables, as in (2.29). The resulting relaxation is (the matrix inequalities are considered elementwise)

$$\begin{array}{ll} \text{maximize} & \frac{1}{4}L \bullet X, \\ \text{subject to} & diag(X) = e, \\ & X + X_1e^T + eX_1^T \;\; \geq \;\; -ee^T, \\ & X - X_1e^T - eX_1^T \;\; \geq \;\; -ee^T, \\ & X - X_1e^T + eX_1^T \;\; \geq \;\; -ee^T, \\ & X = X^T. \end{array} \tag{3.33}$$

Furthermore, we can use both RLT constraints and triangle inequalities

$$\begin{array}{ll}\text{maximize} & \frac{1}{4}L \bullet X,\\ \text{subject to} & diag(X) = e,\\ & l \leq x \leq u,\ X = X^T,\\ & \left.\begin{array}{rcl} X_{ij} + X_{jk} + X_{ik} & \geq & -1,\\ X_{ij} - X_{jk} - X_{ik} & \geq & -1,\\ -X_{ij} - X_{jk} - X_{ik} & \geq & -1,\\ -X_{ij} + X_{jk} - X_{ik} & \geq & -1, \end{array}\right\} i,j,k \in V \\ & \begin{array}{rcl} X + X_1e^T + eX_1^T & \geq & -ee^T,\\ X - X_1e^T - eX_1^T & \geq & -ee^T,\\ X - X_1e^T + eX_1^T & \geq & -ee^T. \end{array}\end{array} \tag{3.34}$$

3.3.6 Doubly nonnegative relaxations

For these relaxations we will need to reformulate max-cut as a problem with binary 0-1 variables. The max-cut problem (3.3) is

$$\begin{array}{ll}\text{maximize} & \frac{1}{4}x^TLx,\\ \text{subject to} & x \in \{-1,1\}^n.\end{array}$$

Using the linear transformation $x = 2y - e$ we obtain a problem with a variable $y \in \{0,1\}^n$. And the objective reduces to

$$\frac{1}{4}x^TLx = \frac{1}{4}(2y-e)^TL(2y-e) = y^TLy - y^TLe + \frac{1}{4}e^TLe = y^TLy,$$

because $Le = 0$ from the definition of L

$$Le = diag(We)e - We = We - We = 0.$$

Hence the binary 0-1 formulation of max-cut is

$$\begin{array}{ll}\text{maximize} & y^TLy,\\ \text{subject to} & y \in \{0,1\}^n.\end{array} \tag{3.35}$$

DNN relaxation

A first DNN relaxation we will obtain from SDP by adding the nonnegative cone constraint, since $y \in \{0,1\}^n$, it holds that $y \geq 0$ and that $Y = yy^T \in \mathbb{N}^n$. Therefore we can relax $Y = yy^T$ as $Y \succeq yy^T$ and $Y \in \mathbb{N}^n$. We will also use the constraint on diagonal entries $Y_{ii} = y_i^2 = y_i$. Hence we get the following relaxation

$$\begin{array}{ll} \text{maximize} & L \bullet Y, \\ \text{subject to} & diag(Y) = y, \\ & Y \succeq yy^T, \\ & Y \in \mathbb{N}^n. \end{array}$$

Which can be easily rewritten, using the Schur complement lemma (see Theorem A.2 in the Appendix) with the doubly nonnegative variable in $\mathbb{S}_+^{n+1} \cap \mathbb{N}^{n+1}$. The resulting relaxation is

$$\begin{array}{ll} \text{maximize} & L \bullet Y, \\ \text{subject to} & diag(Y) = y, \\ & \begin{pmatrix} 1 & y^T \\ y & Y \end{pmatrix} \in \mathbb{S}_+^{n+1} \cap \mathbb{N}^{n+1}. \end{array} \tag{3.36}$$

Stronger DNN relaxation

Following the procedure by Kim and Kojima [22], which we have described in the Section 2.6.1, we fist add the slack variables $u \in \{0,1\}^n$ as a complementary 0-1 variables to y

$$\begin{array}{ll} \text{maximize} & L \bullet yy^T, \\ \text{subject to} & y_i u_i = 0, \ (i = 1, \ldots, n), \\ & y + u = e, \\ & y \geq 0, \quad u \geq 0. \end{array} \tag{3.37}$$

Then introducing the matrix variables $Y = yy^T$, $U = uu^T$ and $W = uy^T$ and relaxaing them we receive the following relaxation

$$\begin{array}{ll} \text{maximize} & L \bullet Y, \\ \text{subject to} & y + u = e, \\ & diag(Y) = y, \quad diag(U) = u, \quad diag(W) = 0, \\ & \begin{pmatrix} 1 & y^T & u^T \\ y & Y & W^T \\ u & W & U \end{pmatrix} \in \mathbb{S}_+^{2n+1} \cap \mathbb{N}^{2n+1}. \end{array} \tag{3.38}$$

This formulation has more variables than (3.36), but is stronger, at least because the variables y are not only nonnegative, but also from nonneqativity of u we have that $y \leq 1$. Furthermore, the semidefinite constraint also provides better links between these variables. In fact, if doubly nonnegative constraint in (3.38) would be replaced by the completely positive constraint, it would be an equivalent formulation of the original problem (3.35). This is due to results by Arima, Kim and Kojima [19].

Lagrangian DNN relaxation

We will derive Lagrangian DNN relaxation for (3.37), using the procedure form Section 2.6.1. Let $A = (-e\ I_n\ I_n)$, $Q_1 = A^T A$,

$$H_0 = \begin{pmatrix} 1 & 0^T & 0^T \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad Q_2 = \begin{pmatrix} 0 & 0^T & 0^T \\ 0 & 0 & I_n \\ 0 & I_n & 0 \end{pmatrix},$$

$$Q_0 = \begin{pmatrix} 0 & 0^T & 0^T \\ 0 & L & 0 \\ 0 & 0 & 0 \end{pmatrix} \quad \text{and} \quad Z = \begin{pmatrix} z_0 & y^T & u^T \\ y & yy^T & yu^T \\ u & uy^T & uu^T \end{pmatrix}.$$

We can enforce $z_0 = 1$ by $Z \bullet H_0 = 1$. Then for $y, u \in \mathbb{R}^n_+$ it holds that $y + u = e$ is equivalent to $Q_1 \bullet Z = 0$ and the complementarity constraints $y_i u_i = 0$ for $i = 1, \ldots, n$ are equivalent to $Q_2 \bullet Z = 0$. Since both matrices Q_1, Q_2 are nonnegative and so is Z, then $Q_1 \bullet Z \geq 0$ and $Q_2 \bullet Z \geq 0$. Therefore two constraints $Q_1 \bullet Z = 0$ and $Q_2 \bullet Z = 0$ can be rewritten as single constraint $H_1 \bullet Z = 0$ for $H_1 = Q_1 + Q_2$.

The structure of Z is equivalent to Z being positive rank 1 matrix

$$Z \in \Gamma^{2n+1} = \{xx^T \mid x \in \mathbb{R}^{2n+1}_+\}.$$

Now we can rewrite the problem (3.37) in he new variable Z as

$$\begin{array}{ll} \text{maximize} & Q_0 \bullet Z, \\ \text{subject to} & H_0 \bullet Z = 1, \quad H_1 \bullet Z = 0, \\ & Z \in \Gamma^{2n+1}. \end{array}$$

Relaxing the cone Γ^{2n+1} with nonnegative cone $\mathbb{S}^n_+ \cap \mathbb{N}^n$ we obtain a DNN relaxation (equivalent with (3.38)). And for chosen λ we obtain following Lagrangian relaxations

$$L^p(\lambda) := \min\{Q_0 \bullet Z - \lambda H_1 \bullet Z \mid H_0 \bullet Z = 1,\ Z \in \mathbb{S}^n_+ \cap \mathbb{N}^n\}, \tag{3.39}$$

$$L^d(\lambda) := \max\{y_0 \mid -Q_0 + \lambda H_1 + y_0 H_0 \in \mathbb{S}^n_+ + \mathbb{N}^n\}. \tag{3.40}$$

Remark 3.41. We have formed a primal dual pair according to (2.40)-(2.41) with $\mathcal{K} = \mathbb{S}^n_+ \cap \mathbb{N}^n$ and $\mathcal{K}^* = \mathbb{S}^n_+ + \mathbb{N}^n$ (See Example A.11 in the Appendix). Since we are maximizing, the signs differ from (2.40)-(2.41).

Chapter 4

Computational comparison

In this section we will try to compare all of the methods we have managed to formulate for max-cut in the Section 3.3.

All of the computations are done in the `Matlab` environment using the `cvx library` with an open source solver `SeDuMi.` Used machine had an Intel Core i7-4710HQ 2.50GHz CPU and 16GB memory. Our code is available on GitHub repository[1] together with this thesis.

We will refer to used methods by following abbreviations.

sdp	sdp rlt	sdp tri	socp1	socp2	socp3	mix1	mix2
(3.12)	(3.16)	(3.12) + (3.17)	(3.19)	(3.25)	(3.27)	(3.28)	(3.30)

mixr r	lp1	lp2	lp3	lp4	dnn1	dnn2	dnn3p	dnn3d
(3.29) given r	(3.31)	(3.32)	(3.33)	(3.34)	(3.36)	(3.38)	(3.39)	(3.40)

Since we are going to compare a long list of methods, we did not manage to optimize each of their implementations, nor solving algorithms. While we are using the `cvx library` with the `SeDuMi` solver, the quality of bounds should not be affected by the flaws of our implementation up to the presented precision. However, the running times are largely affected, since, for each class of methods, we are not using the best-known state of the art algorithms, but only a universal solver, which is more suitable for some classes than for the others. Namely LP and SDP relaxations are expected to have good performance because they are widely used classes. On the other hand, SOCP and DNN relaxations will probably take more time to solve then they would need using a specialized algorithms and efficient implementations.

Remark 4.1. We would like to emphasize especially that our poor time performance of the DNN3p-DNN3d pair compared to SDP is purely due to our implementation. Efficient algorithms for solving this pair of problems have recently been proposed by Arima, Kim, Kojima and Toh [22, 23]. Their reported performance seems to be superior to SDP both in time and quality of the bounds.

[1]`https://github.com/matus-stehlik/conic_relaxations`

In the comparison, we will therefore focus firstly on the quality of the produced bounds and secondly on the running time.

In the first section, we will compare all the relaxation methods using the small ($n = 20$) generated instances of max-cut.

Next we will consider the medium sized max-cut instances ($n \in \{60, 80, 100\}$), for all the methods that we are able to solve in reasonable time (in our implementation).

After this round we will choose only fast methods to compare using the large scale instances ($n \in \{100, 200, \dots, 600\}$).

Last but not least, 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. Additionally we will try to compare the performance of chosen methods in this procedure for max-cut.

4.1 Single bound comparison

As noted above, in the comparison we will focus on the quality of the bounds (relative errors) and the speed (running time) of given methods.

The relative errors for the produced bounds are computed as

$$err = \frac{\theta_{relax} - \hat{\theta}}{\hat{\theta}}, \tag{4.2}$$

where $\hat{\theta}$ is the optimal value of the original problem and θ_{relax} is the computed upper bound (an optimal value of the relaxed problem).

In order to capture both the large diversity of relaxation method classes and the small distinctions between methods of similar performance in a single graph, we are using a logarithmic scale for relative errors and running time.

4.1.1 Small sized instances

All of the mentioned methods were compared using a small generated instnces of max-cut with integer edge weights $1 \leq w_{ij} \leq 50$ and various densities.

For each instance the optimal value was obtained by our branch and bound method.

For a small density 10% most of the produced bounds were exact (see Figure 4.1). As the density grows, the portion of exact bounds decreases. In the other graphs with density 50% very few exact bounds were achieved.

In the figures 4.2, 4.3, we can see that adding the valid RLT inequalities (sdp rlt, lp3, lp4) or triangle inequalities (sdp tri, socp3, lp2, lp4), improves bounds but also increases the run time.

Suprisingly, SOCP relaxations (socp1, socp2, socp3) have very poor performance, at least in our implementation.

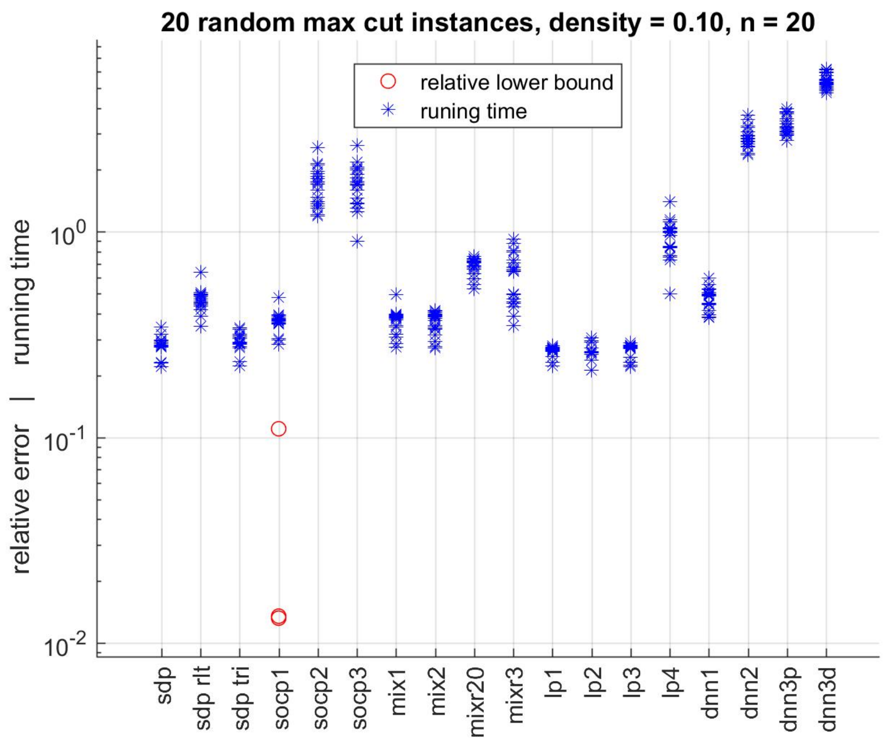


Figure 4.1: Comparison of the relaxation methods for the small sized random generated instances max-cut with density 10% and $n = 20$. On the logarithmic scale we only see nonzero relative errors. All missing red circles represent the exact bounds.

Now lets focus on the changes with the size of the problem. In the figures 4.2, 4.3 one can see that while quality of the bounds stays about the same, the time complexity for some of the methods grows rapidly. Namely the SOCP relaxation with triangle inequalities (socp3) and DNN relaxations (dnn2, dnn3p, dnn3d). The former grows rapidly probably due to large number of triangle inequalities and the latter due to not using an efficient algorithm for DNN. From the practical reasons we are going to exclude these methods from further considerations on medium sized problems.

4.1.2 Medium sized instances

We have chosen all the methods with reasonable running time to compare on the $n = 60$ and $n = 80$ instances. See the Figure 4.4.

An interesting observation is that for some methods (sdp rlt, sdp tri, socp2, lp2, lp4, dnn1) the run times are well separated for $n = 60$ and $n = 80$ instances. On the other hand, the bounds seem to be stable.

Finally we can see in the Figure 4.4 that mixed SOCP-SDP relaxations are getting slightly faster than SDP and that their bounds are quite good, especially mix2 seems to perform very well. Similar to the small instances, SOCP methods (socp1, socp2) do not produce reasonable bounds.

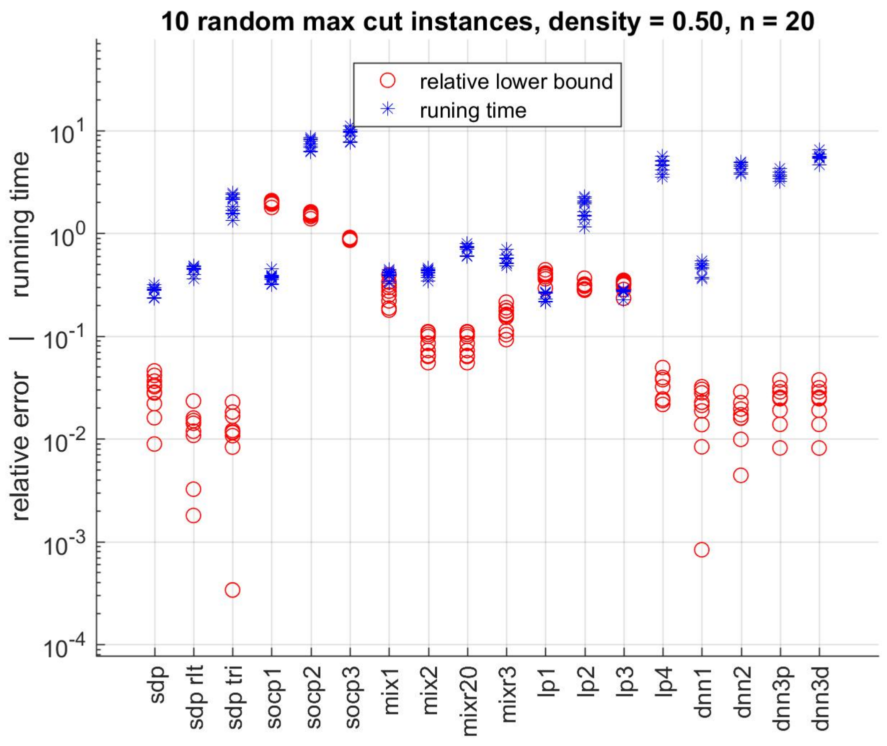


Figure 4.2: Comparison of the relaxation methods for the small sized random generated instances max-cut with density 50% and $n = 20$.

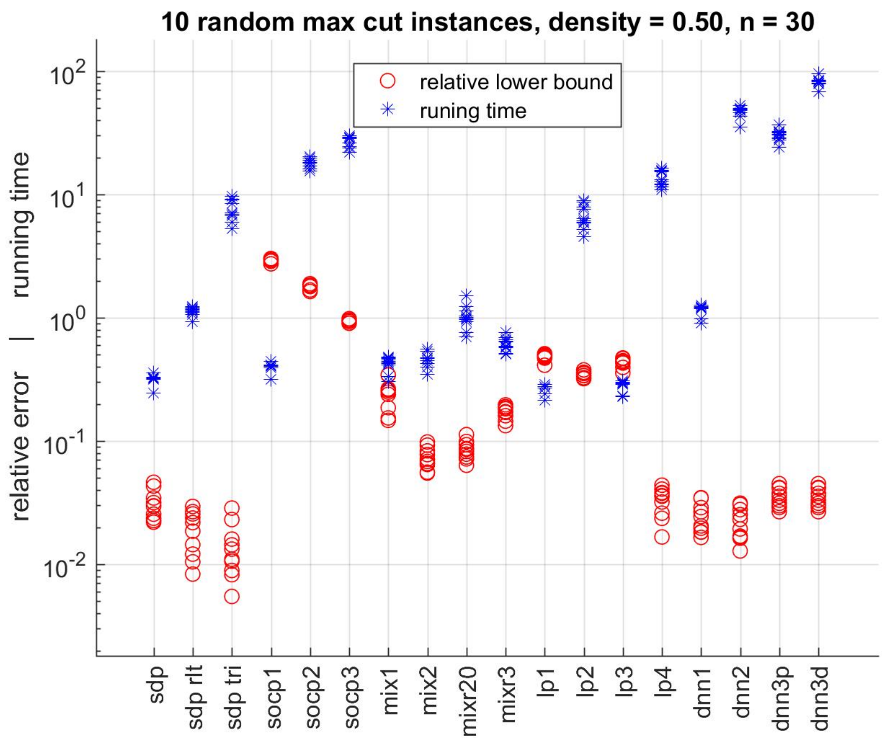


Figure 4.3: Comparison of the relaxation methods for the small sized random generated instances max-cut with density 50% and $n = 30$.

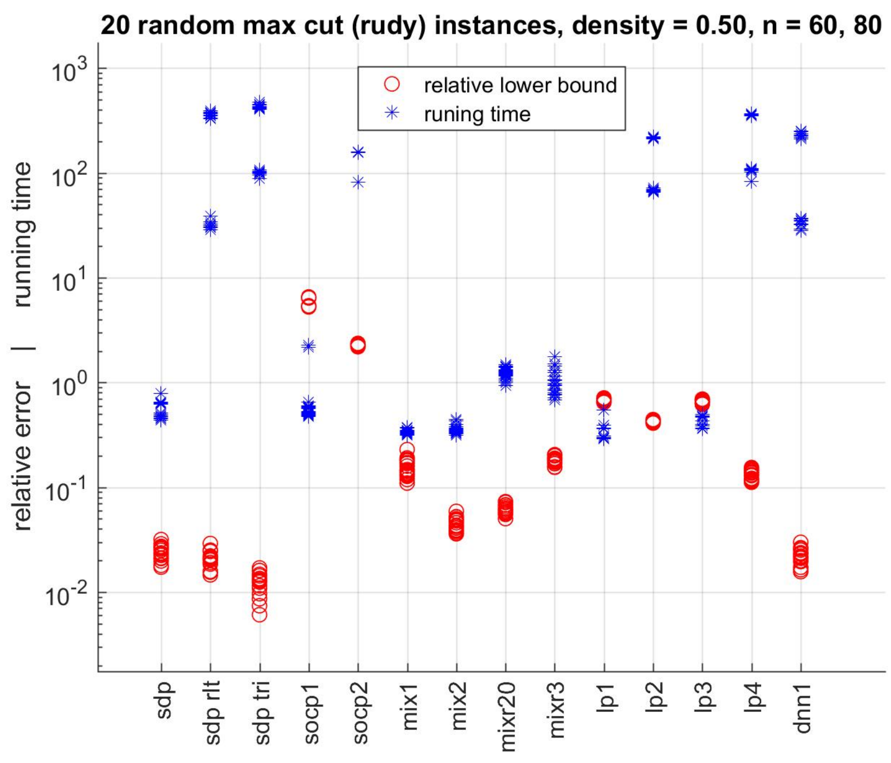


Figure 4.4: Comparison of the relaxation methods on the medium sized max-cut Rudy instances `g_05_60`, `g_05_80`, from the Biq Mac library [39], with density 50%, unit edge weights and $n = 60,\ 80$.

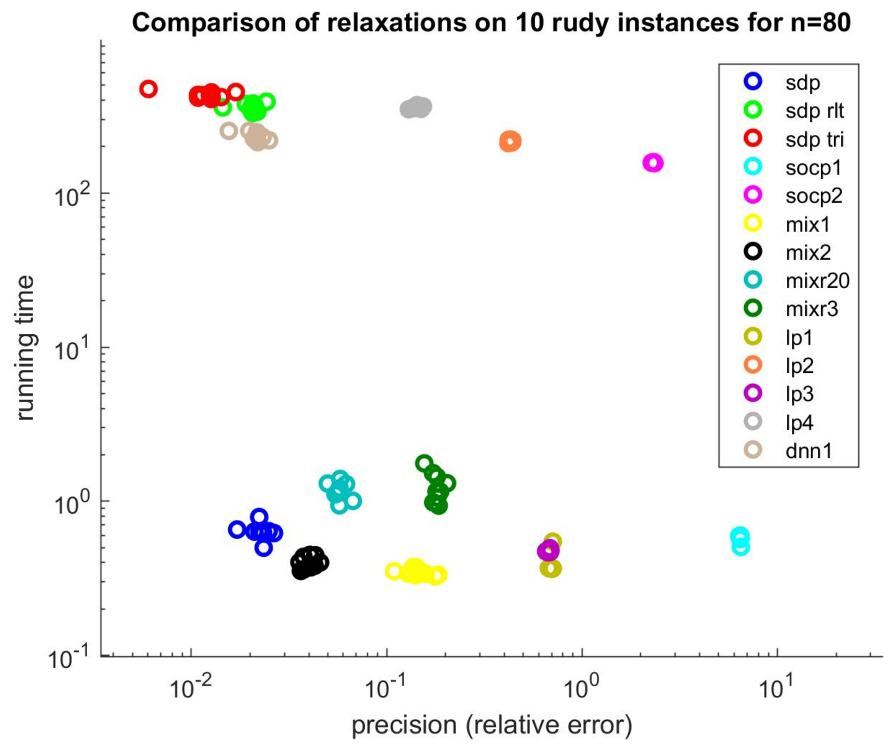


Figure 4.5: A time-precision trade off on rudy instances for $n = 80$.

An interesting comparison is captured in Figure 4.5. One can see the time-precision trade off for the methods close to axes. Also it is obvious that some methods are inferior in both time and precision to others. The empty space for the high precision - medium speed methods can be filled either by including only a portion of the valid inequalities, or as we suggest later, by using the fast relaxation methods in the branch and bound framework.

4.1.3 Large scale instances

In this section we are going to compare only fast methods (sdp, lp1, lp3, mix1, mix2 and mixr for $r = 50, 20, 10, 5$) to see how the performance changes when problems grow large $n = 100, 200, \ldots, 600$.

For this comparison (in the Figures 4.6, 4.7) we did not know the optimal values (it would be almost impossible for us to compute them), hence, we have approximated the relative errors as relative errors against the best obtained bound, which was for each instance the bound produced by SDP relaxation. We will prove later that this relation always holds for SDP and mixed SOCP-SDP relaxation.

In the Figure 4.6 one can see that from the mixed methods the best bounds are produced by mix2 relaxation which has also one of the shortest run times.

We offer also a problem-size breakdown of this experiment in the Figure 4.7 where it is clear that mix2 is superior for all sizes of problems. Moreover, that for all of the methods have more-less stable quality of the bounds and increasing run time for growing n.

4.1.4 Relation of Mix and SDP

In the following, we would like to demonstrate that these mixed SOCP-SDP relaxations are for max-cut never stronger then SDP relaxation.

Let $L = L^- + L^+$ be a splitting of L with $L^- \preceq 0$. We will consider a general mixed SOCP-SDP relaxation of (3.3).

$$\begin{array}{ll} \text{maximize} & \frac{1}{4}\left[X_1^T L^- X_1 + L^+ \bullet X\right], \\ \text{subject to} & diag(X) = e, \\ & X \succeq 0, \end{array} \tag{4.3}$$

where X_1 denotes first column of X. We will refer to this problem as MIX.

Remark 4.4. Notice that all of the mixed methods (mix1, mix2, mixr for any r) are included in this formulation for a certain choice of splitting L^+, L^-. Using the fact that $L^+ \bullet X$ reduces to $Tr(L^+)$ when L^+ is diagonal yields an equivalent formulation.

Theorem 4.5. *Consider an instance of QCQP 2. Denote $\nu(SDP)$, $\nu(Mix)$ the optimal values of the SDP relaxation (3.12) and mixed SOCP-SDP relaxation MIX*

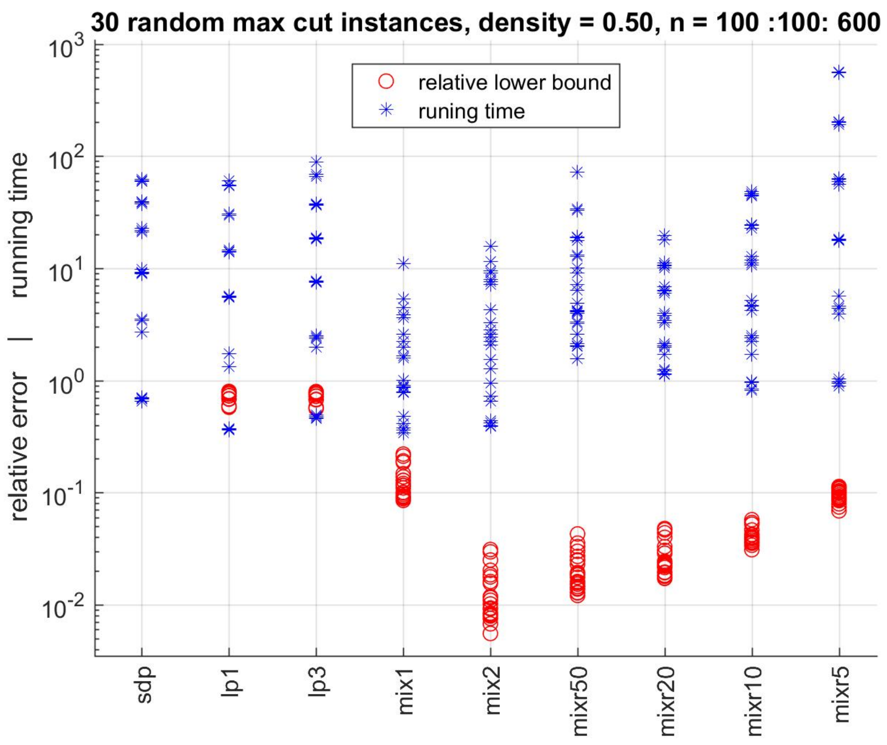


Figure 4.6: Comparison of the relaxation methods on the large max-cut instances with density 50%, integer edge weights between 1 and 50, $n = 100, 200 \ldots, 600$. The relative error is approximated using SDP relaxation as benchmark and computing errors relatively to the bounds produced by sdp.

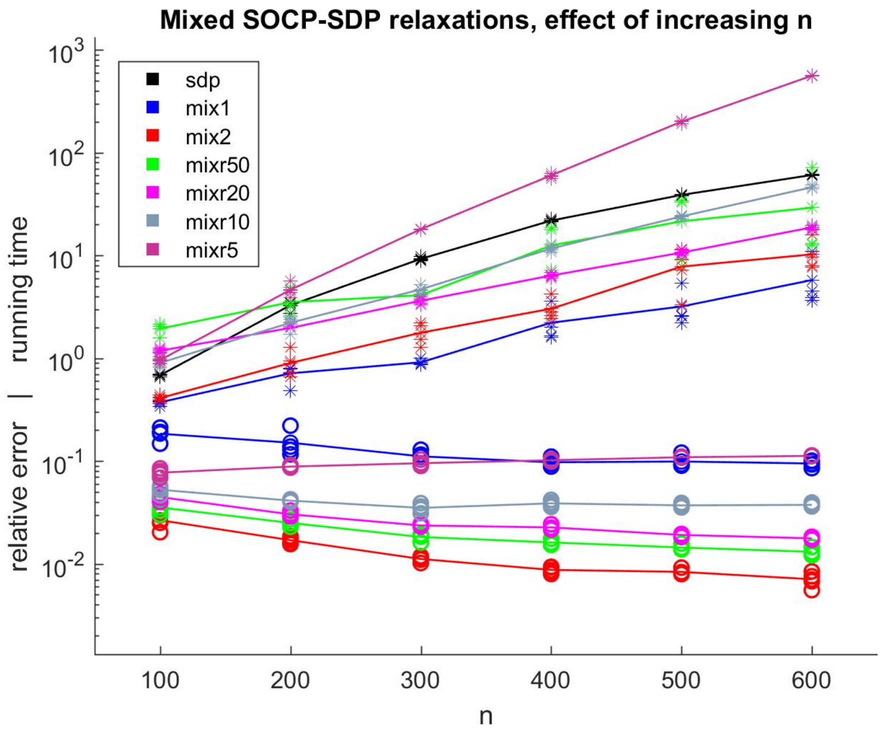


Figure 4.7: Comparison of the mixed relaxation methods on the large max-cut instances with density 50%, integer edge weights between 1 and 50, $n = 100, 200 \ldots, 600$. The relative error is approximated using SDP relaxation as benchmark and computing errors relatively to the bounds produced by sdp. Lines are connecting the mean values for each size and mothod.

(4.3) respectively. It holds that $\nu(SDP) \leq \nu(MIX)$.

Proof. Notice, that both relaxations have same feasible set $F = \{X \in \mathbb{S}^n \mid diag(x) = e,\ X \succeq 0.\}$. We will show that for any $X \in F$ the objective value of SDP is at most the objective value of MIX. Indeed

$$\begin{aligned} L \bullet X - (X_1^T L^- X_1 + L^+ \bullet X) &= L^- \bullet X - X_1^T L^- X_1 \\ &= L^- \bullet (X - X_1 X_1^T) \\ &\leq 0. \end{aligned} \tag{4.6}$$

The inequality hods because $L^- \preceq 0$ and $X - X_1 X_1^T \succeq 0$, since from $X \succeq 0$, using that$X_{11} = 1$, X_1 is the first column of X and Schur complement lemma (see A.2 in the Appendix) we have $X \succeq X_1 X_1^T$. □

Remark 4.7. Using the difference (4.6) expressed in he previous proof, we can bound a a difference of $\nu(MIX) - \nu(SDP)$ in tho following way. Let $\hat{X}, \hat{X}_1$ be the optimal solution of MIX with the objective $\nu(MIX)$. Then $\hat{X}$ is also a feasible solution of SDP with the objective $\nu_{SDP}(\hat{X})$ such that

$$\nu(MIX) \geq \nu(SDP) \geq \nu_{SDP}(\hat{X}) \geq \nu(MIX) + \frac{1}{4} L^- \bullet (\hat{X} - \hat{X}_1 \hat{X}_1^T).$$

Hence, if $|L^- \bullet (\hat{X} - \hat{X}_1 \hat{X}_1^T)|$ is small, the the bound of MIX is guaranteed to be very close to bound produced by SDP.

4.2 Branch and bound

We will first describe the general idea of the branch and bound algorithms. Suppose we are dealing with a 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 $x_{-l} \in \{0, 1\}^n$.

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 a chosen subproblem. It is often extracted from the solution of the relaxed problem using a projection or a 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.

SDP				mix			
opt	topIter	iter	time	opt	topIter	iter	time
3093	4	72	38.978	3093	4	371	241.51
3342	3	70	44.229	3342	5	854	710.19
3262	1	108	54.709	3262	4	723	493.06
3330	54	130	69.598	3330	52	1000	750.6
3279	130	171	107.22	3279	1	1306	1099.7
3216	190	237	151.78	3216	947	1626	1387.1
3293	143	175	117.01	3293	215	1155	976.52
3335	4	44	29.234	3335	4	392	327.49
3322	20	123	67.561	3322	409	764	506.83
3382	44	117	75.352	3382	117	865	734.63
2866	170	478	315.12	2861	90	3000	2535.4

Table 4.1: Table of branch and bound solution instances for $n = 40$, density 50%. Columns: opt - optimal value or the best solution found, topIter - a number of iteration when opt was found, iter - total number of iterations, time - total time. A maxumum number of iterations was set to 3000 (the last instance of mix did not finish)

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.

4.2.1 Our branch and bound framework

We will be branching the problem tree by fixing the variables x_i. At the beginning the only fixed variable is $x_1 = 1$. This is without loss of generality, because if there is an optimal solution $\hat{x}$ then $-\hat{x}$ having the same objective value is also an optimal solution.

Problem formulations

Each problem solved during the branch and bound procedure will be represented by sets of indices fixed to 1 and -1:

$$P = \{i \in V \mid x_i = 1 \text{ is fixed}\}, \qquad N = \{i \in V \mid x_i = -1 \text{ is fixed}\},$$

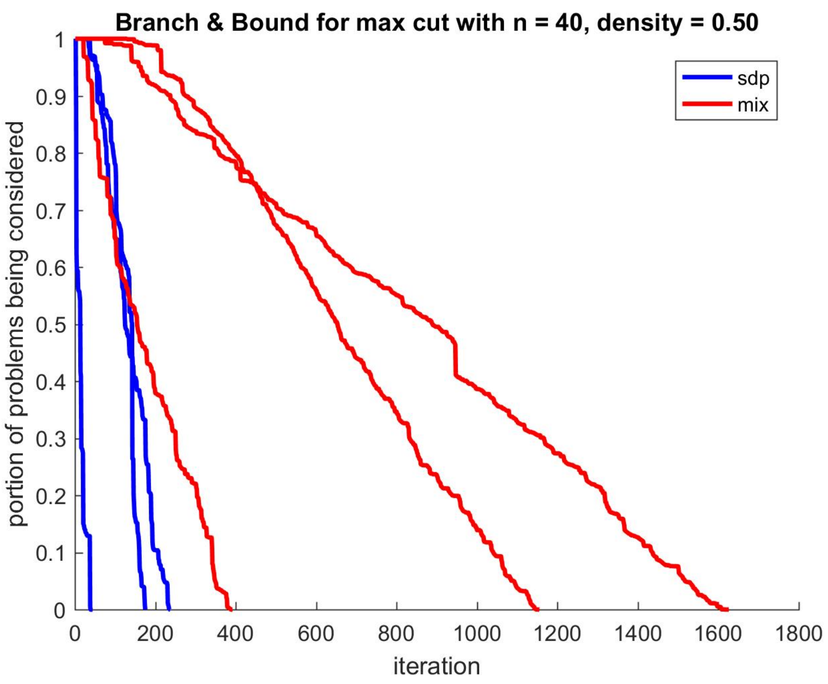


Figure 4.8: Branch and bound comparison of the relaxation methods SDP and mixed SOCP-SDP relaxation, on 3 random generated instances.

with $P \cap N = \emptyset$. Let us denote the set of known indices $K = P \cup N$ and set of unknown indices $U = V \setminus K$.

Now the objective $x^T Lx/4$ of the problem (3.3) with fixed indices reduces to

$$\begin{aligned} \frac{1}{4}x^T Lx &= \frac{1}{4}(x_K^T, x_U^T)\begin{pmatrix} L_{KK} & L_{KU} \\ L_{KU} & L_{UU} \end{pmatrix}\begin{pmatrix} x_K \\ x_U \end{pmatrix} \\ &= \frac{1}{4}\left[x_U^T L_{UU} x_U + 2x_K^T L_{KU} x_U + x_K^T L_{KK} x_K\right]. \end{aligned}$$

The resulting problem of the size $n - k$ with variable $x_U \in \{0,1\}^n$ is

$$\begin{array}{ll} \text{maximize} & \frac{1}{4}\left[x_U^T L_{UU} x_U + 2x_K^T L_{KU} x_U + x_K^T L_{KK} x_K\right], \\ \text{subject to} & x_U \in \{0,1\}^n. \end{array}$$

It has following SDP relaxation

$$\begin{array}{ll} \text{maximize} & \frac{1}{4}\left[L_{UU} \bullet X_{UU} + 2x_K^T L_{KU} x_U + x_K^T L_{KK} x_K\right], \\ \text{subject to} & diag(X_{UU}) = 1, \\ & X_{UU} \succeq x_U x_U^T. \end{array}$$

And following mixed SOCP-SDP relaxation (mix2)

$$\begin{array}{ll} \text{maximize} & \frac{1}{4}\left[\ x_U^T(\bar{D}_{LU} - \lambda_{max}(\bar{D}_{LU})I)x_U + 2x_K^T L_{KU} x_U + \right. \\ & \left. \quad + x_K^T L_{KK} x_K + e^T D_{LU} + \lambda_{max}(\bar{D}_{LU})(n-k)\ \right], \\ \text{subject to} & x_i^2 \leq 1,\ (i = 1, \ldots, n), \end{array}$$

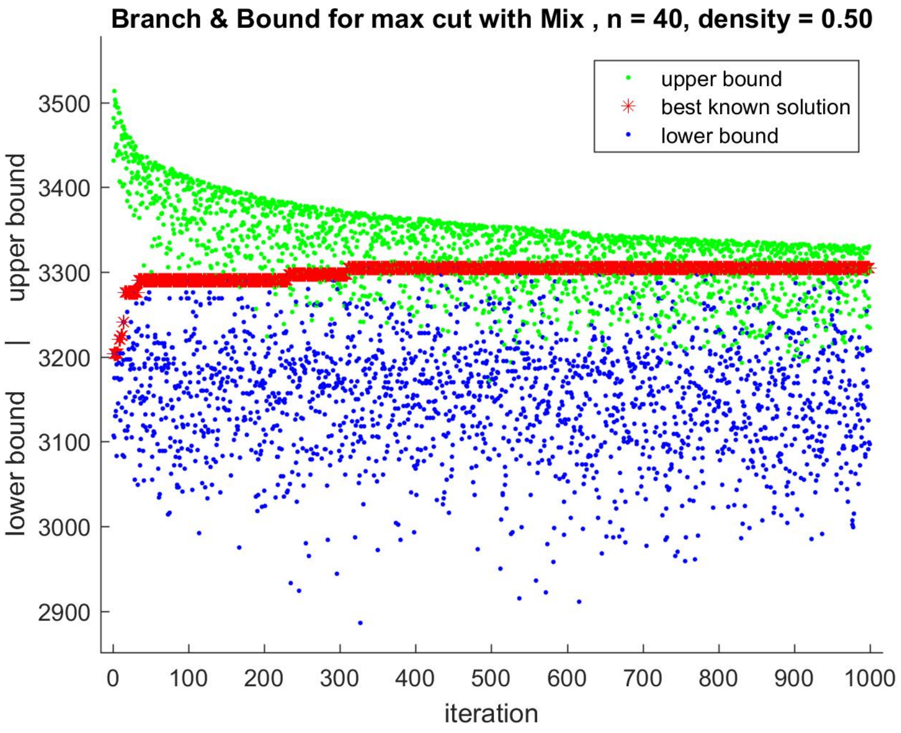


Figure 4.9: Branch and bound for mixed SOCP-SDP relaxation.

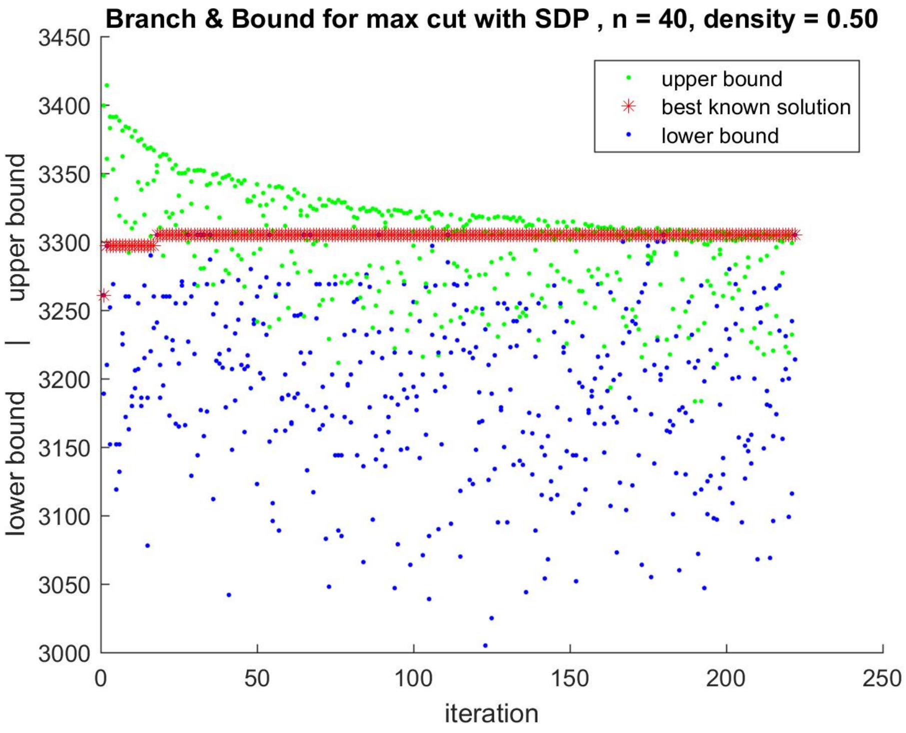


Figure 4.10: Branch and bound for SDP relaxation.

where D_{LU} is the $n-k \times n-k$ diagonal matrix with diagonal entries of L_{UU}, $\bar{D}_{LU} = L_{UU} - D_{LU}$ is its complement and $\lambda_{max}(\bar{D}_{LU})$ denotes the largest positive eigenvalue of $\bar{D}_{LU}$.

Branching

For the branching we have tried many approaches, of which the most reasonable seemed to be fixing either vertices with large degrees (in the remaining graph) to sparsify the residual graph or the vertices with the greatest contribution to the scalar term $x_K^T L_{KK} x_K$ after being fixed.

Both of these approaches are heuristics, first one aims to simplify the graph, the second one tries to secure large part of the objective as soon as possible. That suggest a compromise of these may be a good idea. That is in each node we will randomly decide either with probability p_d to branch by the degree and with probability $1-p_d$ to branch by the weight of contribution.

It seem that for dense graphs larger p_d works better, on the other hand for sparse graphs, smaller p_d gives better results.

Producing lower bounds

For the SDP relaxation we will use the rounding algorithm by Goemans and Williamson [37] (see the Section 3.2). For the mixed relaxation we will use simple rounding of x to $sign(x)$.

Remark 4.8. We have tried to develop another rounding procedures for mix2. Specifically, we have considered creating a matrix $X \succeq 0$ with $diag(X) = e$ such that x would be its first column, in order to use the GW-algorithm for SDP with guaranteed 0.878 performance. We tried to construct X as a feasible soulution for SDP with objective as close to the objective of mix as possible. To keep the most information from x a following structure of X was expected $X = xx^T + diag(u) + \bar{U}$, where $diag(\bar{U}) = 0$. The diagonal part is determined by requiring $diag(X) = e$. On the other hand, finding $\bar{U}$ such that difference of the objectives (4.6) is minimal and $xx^T + diag(u) + \bar{U} = X \succeq 0$ is an SDP in variable $\bar{U}$./

Rounding upper bounds

In our experiments we will only test instances with integer edge weights. Therefore, the optimal value is also expected to be integer. In our experiments we will always improve the upper bounds by rounding them down.

4.2.2 Branch and bound comparison

In this section we are going to comapre mix2 and sdp relaxations in th branch and bound procedure. In the table 4.1 one can see that solving the max cut to optimality takes several times longer with mixed SOCP-SDP than with SDP bounds. However, one can see that optimal solution is usually found quite soon. It is the proof of optimality that takes most of the time.

Before we have seen that differences in bounds are small in relative, but in the Figures 4.9, 4.10 it is clear those small relative differences are significant for proving the optimality.

Lastly the Figure 4.8 shows how the branching procedure manages to exclude problems in time.

An interesting observation is that usually the optimum is found after a fraction of time needed to complete the BnB procedure. Therefore few iterations of BnB can be used to strengthen the relaxation. It can improve both the upper bound as a maximum of the upper bounds of all considered subproblems, and the lower bound as the best found feasible solution. One can use heuristics for ordering the subproblems in BnB with aim to focus more on the former (the high upper bounds are computed first) or the latter (branches where the best feasible solutions were found are explored first). This approach offers further flexibility in choosing the time-precision trade off of the relaxation techniques.

Conclusion

This thesis surveyed the relaxation methods, mostly for QCQP. We introduced both well known approaches for constructing relaxations (such as LP and SDP relaxations), as well as approaches that are unfamiliar (SOCP and mixed relaxations) or recent (Lasserre hierarchy, DNN relaxations, CP and CPP reformulations). Throughout this thesis we discussed techniques for strengthening the relaxations gain the precision - adding valid inequalities (RLT, PSD-cuts, triangle inequalities for max-cut). But also techniques for decreasing the problem size and running time (SOCP and mixed SOCP-SDP relaxations, Lagrangian DNN primal-dual pair)

In the first chapter we gave organized introduction to conic programming classes showing key relations between them. We also highlighted their common structure by associating them with the general cone programming class. We have further used this class to derive dual problems for all conic classes at once.

The second chapter offers extensive overview of relaxation techniques. We have gathered many approaches of relaxing the quadratic problems by conic optimization classes and arranged them systematically, showing the motivations for the techniques and providing links between them.

In the third chapter we profoundly explore the application of the relaxation methods to the maximum cut problem. We formulate 16 relaxations, including their adaptations strengthened with valid inequalities.

These methods were compared by numerical experiments in the fourth chapter, which confirmed that SDP relaxation seems to be the most reasonable compromise, both in speed and quality of the produced bounds. In addition for a computational cost, SDP can be strengthened by valid inequalities. On the other hand, also the mixed SOCP-SDP methods proved to be reasonable compromise offering both quality bounds and exceptional time performance. At the end we suggest a possibility of using the branch and bound procedure for strengthening the relaxations.

In the appendix we cover basic properties of cones and include examples showing the important properties of the cones used throughout the thesis.

We believe that this thesis may serve as comprehensive guide through the field of relaxation methods for problems of quadratic programming.

Appendix A

A.1 Schur complement

We will first define the Moore-Penrose pseudoinverse which we will use in following theorem.

Definition A.1 (Moore-Penrose pseudoinverse)**.** For a matrix $A \in \mathbb{R}^{m\times n}$ a pseudoinverse of A is defined as a matrix $A^\dagger \in \mathbb{R}^{n\times n}$ satisfying all of the following

$$\begin{aligned} AA^\dagger A &= A, \\ A^\dagger AA^\dagger &= A, \\ (AA^\dagger)^T &= AA^\dagger, \\ (A^\dagger A)^T &= A^\dagger A. \end{aligned}$$

Theorem A.2 (Schur complement lemma [1, 42, 43])**.** *Let M be a symmetric matrix of the form*

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

Then $M \succeq 0$ (M is positive semidefinite) if and only if

$$A \succeq 0, \quad \mathcal{R}(B) \subseteq \mathcal{R}(A), \quad C - B^T A^\dagger B \succeq 0.$$

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

Proof. A main idea of the proof is explained in [1]. For a rigorous proof see any of [42, 43].

A.2 Cones

This Section is focused on providing introduction to the theory of cones and overview of basic claims and definitions. Majority of the content in this section is from [10]. At 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.3 (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.4 (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 any line, i.e. if $\pm x \in \mathcal{K}$, then $x = 0$).

Definition A.5 (Dual cone)**.** For any cone $\mathcal{K}$ we define its dual cone $\mathcal{K}^*$ as a set

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

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

Proposition A.6 ([10])**.** *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.7. *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 sufficient 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 suffices to prove that $\mathcal{K}^* \cap (-\mathcal{K}*) = \{0\}$. Let $y \in \mathcal{K}^* \cap (-\mathcal{K}*)$, then $x^T y = 0$ for all $x \in \mathcal{K}$. Since $int\ \mathcal{K} \neq \emptyset$, there exist open ball $B \subset \mathcal{K}$. By 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 $0 \notin conv(S^n \cap \bar{\mathcal{K}})$. By contradiction, if $0 \in conv(S^n \cap \bar{\mathcal{K}})$ then there are some $x^1, \ldots, x^m \in \left(S^n \cap \bar{\mathcal{K}}\right)$ 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

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

This is a contradiction, because $\bar{\mathcal{K}}$ is pointed, hence $0 \notin conv(S^n \cap \bar{\mathcal{K}})$. Since $conv(S^n \cap \bar{\mathcal{K}})$ is closed and convex, by separating hyperplane theorem (see [1]) there exists nonzero $a \in \mathbb{R}^n$ and $b \in \mathbb{R}$ such that $a^T x \geq b$ for any $x \in (S^n \cap \bar{\mathcal{K}})$ and $a^T 0 < b$. Therefore $b > 0$ and for any nonzero $y \in \bar{\mathcal{K}}$, since the $y/\|y\| \in (S^n \cap \bar{\mathcal{K}})$ 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

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

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.8. *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 [1]), i.e. there is an $a \in \mathbb{R}^n$, such that

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

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 $\mathcal{K}^{**} = \mathcal{K}$. □

A.3 Examples

Example A.9 (Closed convex cones)**.** The following sets are closed convex cones

1. Nonnegative orthant $\mathbb{R}^n_+$,
2. Nonnegative cone
$$\mathbb{N}^n = \{M \in \mathbb{R}^{n\times n} \mid M_{ij} \geq 0,\ \forall i,j = 1,\dots,n\},$$
3. Second order cone
$$\mathbb{Q}^n = \{x \in \mathbb{R}^n \mid x = (x_0,\bar{x}) \in \mathbb{R}\times\mathbb{R}^{n-1}, \|\bar{x}\|_2 \leq x_0\},$$
4. Set of symmetric positive semidefinite matrices
$$\mathbb{S}^n_+ = \{M \in \mathbb{S}^n \mid x^TMx \geq 0,\ \forall x \in \mathbb{R}^n\},$$
5. Set of copositive matrices
$$\mathbb{C}^n = \{M \in \mathbb{S}^n \mid x^TMx \geq 0,\ \forall x \in \mathbb{R}^n_+\},$$
6. Set of completely positive matrices
$$\mathbb{P}^n = \{M \in \mathbb{S}^n \mid M = \sum_{i=1}^{l} x^i x^{iT} \ \text{ where } x^i \in \mathbb{R}^n_+ \ (i = 1,\dots,n)\}.$$

Proof. 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. – 2. Follows simply from the definition.

3. (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
$$\begin{aligned}\theta_1 x + \theta_2 y &= (\theta_1 x_0 + \theta_2 y_0,\ \theta_1\bar{x} + \theta_2\bar{y}),\\ \|\theta_1\bar{x} + \theta_2\bar{y}\|_2 &\leq \|\theta_1\bar{x}\|_2 + \|\theta_2\bar{y}\|_2 \leq \theta_1 x_0 + \theta_2 y_0.\end{aligned}$$

4. (It is a cone) Let $M \in \mathbb{S}^n_+$, that is $x^TMx \geq 0\ \forall x \in \mathbb{R}^n$ and $\theta \geq 0$. Then
$$x^T(\theta M)x = \theta(x^TMx) \geq 0\ \forall x \in \mathbb{R}^n \ \Leftrightarrow\ (\theta M) \in \mathbb{S}^n_+.$$
(It is convex) Let $M_1, M_2 \in \mathbb{S}^n_+$, and $\theta_1,\theta_2 \geq 0$. Then
$$x^T(\theta_1M_1+\theta_2M_2)x = x^T\theta_1M_1x+x^T\theta_2M_2x \geq 0\ \forall x \in \mathbb{R}^n \ \Leftrightarrow\ (\theta_1M_1+\theta_2M_2) \in \mathbb{S}^n_+.$$

5. Analogically as for $\mathbb{S}^n_+$, only $\mathbb{R}^n$ is replaced by $\mathbb{R}^n_+$.

6. (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, $i = 1, \dots, l$ 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, ($i = 1, \dots, l_1$ and $j = 1, \dots, l_2$) 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.10 (Proper cones)**.** Following cones are proper

1. $\mathbb{R}^n_+$, 2. $\mathbb{N}^n$, 3. $\mathbb{Q}^n$, 4. $\mathbb{S}^n_+$, 5. $\mathbb{C}^n$, 6. $\mathbb{P}^n$.

Proof. Since we have shown that all of them are closed convex cones (see Example A.9), we only need to prove that they are pointed and have nonempty interiors. The first three 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).

4. 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.

5. The $\mathbb{C}^n$ is closed, convex cone (see Example A.9). 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.

6. From the Corrolary A.7 since $\mathbb{C}^n$ is a proper cone, its dual $\mathbb{P}^n$ is also proper.

□

Example A.11 (Dual cones)**.** The following properties about dual cones holds

1. $\mathbb{R}^n_+$ is a selfdual cone,
2. $\mathbb{N}^n_+$ is a selfdual cone,
3. $\mathbb{Q}^n$ is a selfdual cone,
4. $\mathbb{S}^n_+$ is a selfdual cone,
5. $(\mathbb{C}^n)^* = \mathbb{P}^n$ and $(\mathbb{P}^n)^* = \mathbb{C}^n$,
6. $(\mathbb{S}^n_+ \cap \mathbb{N}^n)^* = \mathbb{S}^n_+ + \mathbb{N}^n$.

Proof. 1. – 2. This is an easy observation.

3. 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.

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

5. First we will show $(\mathbb{P}^n)^* = \mathbb{C}^n$. From the definition, using the inner product $Tr(M^T P) = M \bullet P$, 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 $x^T Mx < 0$. Indeed, the $xx^T \in \mathbb{P}^n$ then excludes M from $(\mathbb{P}^n)^*$ since $M \bullet xx^T = x^T Mx < 0$.

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.

6. By definition

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

Since both $\mathbb{S}^n_+$, $\mathbb{N}^n$ are selfdual, it is clear that $(\mathbb{S}^n_+ + \mathbb{N}^n) \subset (\mathbb{S}^n_+ \cap \mathbb{N}^n)^*$. Next also it is easy to see that

$$(\mathbb{S}^n_+ + \mathbb{N}^n)^* \subset (\mathbb{S}^n_+ \cap \mathbb{N}^n), \tag{A.12}$$

because when $M \notin \mathbb{S}^n_+$ then there is $S \in \mathbb{S}^n_+$ such that $M \bullet S < 0$. Also if $M \notin \mathbb{N}^n$ then there is $S \in \mathbb{N}^n$ such that $M \bullet S < 0$. So whe $M \notin (\mathbb{S}^n_+ \cap \mathbb{N}^n)$ there is a matrix $S \in (\mathbb{S}^n_+ + \mathbb{N}^n)$ such that $S \bullet M < 0$, excluding M from $(\mathbb{S}^n_+ + \mathbb{N}^n)^*$.

For any cones $K_1 \subset K_2$ it holds that $K_2^* \subset K_1^*$, therefore from (A.12) we have

$$(\mathbb{S}^n_+ \cap \mathbb{N}^n)^* \subset (\mathbb{S}^n_+ + \mathbb{N}^n)^{**} = (\mathbb{S}^n_+ + \mathbb{N}^n).$$

The last equality holds from Proposition A.8, because $(\mathbb{S}^n_+ + \mathbb{N}^n)$ is closed cone. Thus we have proven both inclusions.

$\square$

References

[1] S. Boyd, L. Vandenberghe, *Convex Optimization.* Cambridge University Press 2004, ISBN 978-0-521-83378-3, seventh printing with corrections, 2009

[2] L. Tuncel, *Polyhedral and Semidefinite Programming Methods in Combinatorial Optimization.* Fields Institute Monograph Series, AMS, Vol.FIM-27, 2010

[3] L. El Ghaoui, *Introduction to Convex Optimization.* A course EE 227A website on UC Berkeley, Available online: `https://inst.eecs.berkeley.edu/~ee227a/fa10/login/index.html`

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

[5] S. Burer, S.Kim, M. Kojima, *Faster, but Weaker, Relaxations for Quadratically Constrained Quadratic Programs.* Computational Optimization and Applications, Vol.59 (1), pp 27-45, 2014

[6] S. Kim, M. Kojima, *Exact Solutions of Some Nonconvex Quadratic Optimization Problems via SDP and SOCP Relaxations.*Computational Optimization and Applications, Vol.26 (2), pp 143-154, 2003

[7] S. Kim, M. Kojima, *Second Order Cone Programming Relaxations of Nonconvex Quadratic Optimization Problems.* Optimization Methods and Software, Vol.15 (3-4), pp 201-224, 2001

[8] S. Kim, M. Kojima, M. Yamashita, *Second Order Cone Programming Relaxation of a Positive Semidefinite Constraint.* Optimization Methods and Software, Vol.18 (5), pp 535-541, 2003

[9] F. Alizadeh, D. Goldfarb, *Second-order cone programming.* Mathematical Programming, Vol.95 (1), pp 3-51, 2003

[10] K. Pokorná, *Second order cone programing.* Master's thesis, Comenius Univeristy, Bratislava, 2013

[11] M. S. Lobo, L. Vandenberghe, S. Boyd, H. Lebret, *Applications of second-order cone programming.* Linear Algebra and its Applications, Vol.284 (1-3), pp 193-228, 1998

[12] A. Qualizza , P. Belotti, F. Margot, *Linear Programming Relaxations of Quadratically Constrained Quadratic Programs.* IMA Volume Series, 154, 2010

[13] K. M. Anstreicher, *Semidefinite Programming versus the Reformulation Linearization Technique for Nonconvex Quadratically Constrained Quadratic Programming.* Journal of Global Optimization, Vol.43 (2), pp 471-484, 2009

[14] H. D. Sherali, W. P. Adams, *A reformulation-linearization technique for solving discrete and continuous nonconvex problems.* Nonconvex Optimization and Its Applications, Vol.31, 1999

[15] H.D. Sherali, B.M.P. Fraticelli, *Enhancing RLT relaxations via a new class of semidefinite cuts.* Journal of Global Optimization, Vol.22 (1), pp 233-261, 2002

[16] X. Zheng, X. Sun, D. Li, *Nonconvex quadratically constrained quadratic programming: best d.c. decompositions and their sdp representations.* Journal of Global Optimization, Vol.50 (4), pp 695-712, 2011

[17] B. Kocuk, S. S. Dey, X. A. Sun, *Inexactness of SDP Relaxation and Valid Inequalities for Optimal Power Flow.* IEEE Transactions on Power Systems, Vol.31, pp 642-651, 2016

[18] H. Dong, J. Linderoth, *On Valid Inequalities for Quadratic Programming with Continuous Variables and Binary Indicators.* Integer Programming and Combinatorial Optimization, Vol.7801 of the series Lecture Notes in Computer Science, pp 169-180, 2013

[19] N. Arima, S. Kim, M. Kojima, *A quadratically constrained quadratic optimization model for completely positive cone programming.* SIAM Journal on Optimization, Vol.23 (4), pp 2320-2340, 2013

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

[21] G. Eichfelder, J. Povh *On the set-semidefinite representation of nonconvex quadratic programs over arbitrary feasible sets.* Optimization Letters, Vol.7 (6), pp 1373-1386, 2013

[22] 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, Vol.156 (1), pp 161-187, 2016

[23] 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, February 2016

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

[25] K. G. Murty, S. N. Kabadi, *Some NP-complete problems in quadratic and non-linear programming.* Mathematical Programming, Vol.39 (2), pp 117–129, 1987

[26] E. Balas, S. Ceria, G. Cornuéjols, *A lift-and-project cutting plane algorithm for mixed 0-1 programs.* Mathematical Programming, Vol.58 (1), pp 295-324, 1993.

[27] L. Lovász, A. Schrijver, *Cones of matrices and set-functions and 0-1 optimization.* SIAM Journal on Optimization, Vol.1 (2), pp 166–190, 1991

[28] H. Sherali, W. Adams. *A hierarchy of relaxation between the continuous and convex hull representations.* SIAM Journal on Discrete Mathematics, Vol.3 (3), pp 411–430, 1990.

[29] J. Lasserre. *An explicit exact SDP relaxation for nonlinear 0-1 programs.* Integer Programming and Combinatorial Optimization Vol.2081 of the series Lecture Notes in Computer Science, pp 293–303, 2001.

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

[31] M. Laurent, *A Comparison of the Sherali-Adams, Lovász-Schrijver and Lasserre Relaxations for 0 - 1 Programming.* Mathematics of Operations Research, Vol.28 (3), pp 470-496, 2001

[32] 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`

[33] S. Kim, M. Kojima, *Binary Quadratic Optimization Problems That Are Difficult to Solve by Conic Relaxations.* Research Report B-481, Department of Mathematical and Computing Sciences, Tokyo Institute of Technology, July 2015

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

[35] U. Raber, *A Simplicial Branch-and-Bound Method for Solving Nonconvex All-Quadratic Programs.* Journal of Global Optimization, Vol.13 (4), pp 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.* Mathematical Programming Computation, Vol.4 (3), pp 275-306, 2012

[37] M. X. Goemans, D. P. Williamson, *Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming.* Journal of the ACM, Vol.42 (6), pp 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, Vol.46 (2), pp 164-177, 2003

[39] A. Wiegele, *Biq Mac Library - A collection of Max-Cut and quadratic 0-1 programming instances of medium size.* 2007
Available online: `http://www.biqmac.uni-klu.ac.at/biqmaclib.pdf`

[40] P. Berman, M. Karpinski, *On Some Tighter Inapproximability Results.* Automata, Languages and Programming Vol.1644 of the series Lecture Notes in Computer Science, pp 200-209 1999

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

[42] F. Zhang, *The Schur Complement and Its Applications.* Numerical Methods and Algorithms Vol.4 2005

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

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