Relations between Semidefinite, Copositive, Semi-infinite and Integer Programming

Author: Faizan Ahmed

Supervisor: Dr. Georg Still

Master Thesis

University of Twente
the Netherlands

May 2010
Relations between Semidefinite, Copositive, Semi-infinite and Integer programming

A thesis submitted to the faculty of Electrical Engineering, Mathematics and Computer Science, University of Twente, the Netherlands in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCES IN APPLIED MATHEMATICS

with specialization

INDUSTRIAL ENGINEERING AND OPERATIONS RESEARCH

DEPARTMENT OF APPLIED MATHEMATICS,

UNIVERSITY OF TWENTE
THE NETHERLANDS

May 2010
Summary

Mathematical programming is a vast area of research in mathematics. On the basis of special structure this field can be further classified into many other, not necessarily completely distinct, classes. In this thesis we will focus on two classes, namely Cone Programming and Semi-infinite Programming.

Semi-infinite programming represents optimization problems with infinite many constraints and finitely many variables. This field emerged in 1924, but the name semi-infinite programming was coined in 1965.

Cone programming is the class of problems in which the variable should belong to a certain cone. The most interesting application of cone programming is cone programming relaxation which has numerous example in combinatorial optimization and other branches of science and mathematics. The most popular and well known cone programs are semidefinite programs. These programs got popularity due to there huge application in combinatorial optimization. Another class of cone programing is copositive programing. Copositive programming has recently gained attention of researchers for their application to solve hard combinatorial optimization problems. Our main focus in this thesis will be on copositive programming.

Another problem of interest is to analyze the way how we can represent these different classes in terms of each other. We will consider the restrictions and benefits we will obtain for these kind of representations. Normally these kind of representations helps to use algorithms available for one class, for the solution/approximation or finding good bounds for other classes of problems.

Eigenvalue optimization can be seen as the building block for the development of semidefinite programming. In this thesis we will investigate this relationship to answer the question whether one can solve semidefinite program by formulating it as an equivalent eigenvalue optimization with the aid of semi-infinite programming.

In summary, SIP and SDP are old and well studied problems and copositive programing is a new area of research. Moreover there are some relationships among copositive, semidefinite and semi-infinite programming. So in this thesis we will focus on these three problems,

1. Survey of Copositive programming and its application to solve integer programs.
2. Semi-infinite representation of copositive and semidefinite programming
3.2.3 Applications of SDP .................................................. 32

4 Copositive Programming (CP) ........................................ 34
  4.1 Copositive Programming ........................................... 34
  4.2 Applications ......................................................... 35
    4.2.1 Quadratic Programming ...................................... 36
    4.2.2 Quadratically Constrained Quadratic Program .......... 38
    4.2.3 Stable Set Problem ........................................... 41
    4.2.4 Graph Partitioning .......................................... 44
    4.2.5 Quadratic Assignment Problem (QAP) .................... 47
    4.2.6 Mixed Binary Quadratic Programming .................... 50
  4.3 Algorithm .......................................................... 58
    4.3.1 Approximation Hierarchy Based Methods ................. 58
    4.3.2 Feasible Descent Method .................................. 60
    4.3.3 $\epsilon$-Approximation Algorithm ....................... 61

5 Semi-Infinite Programming (SIP) Representation of CP and SDP 64
  5.1 Semi-infinite Programming (SIP) ............................... 64
    5.1.1 Linear Semi-infinite Program (LSIP) ..................... 66
  5.2 SIP Representation of CP ....................................... 70
  5.3 SIP Representation of SDP ..................................... 73
  5.4 SIP Solution Approach for SDP ................................ 75

6 Conclusion and Future Work ......................................... 79
  6.1 Conclusion ........................................................ 79
  6.2 Future Work ...................................................... 80

References .............................................................. 83
Chapter 1

Introduction and Literature Review

1.1 Introduction

Mathematical programming represents, the class of problems in which we maximize/minimize some function with respect to some side conditions called constraints. This area of mathematics is further subdivided into classes of convex and non-convex programming.

Convex programming problems are considered less hard comparative to non-convex programming. Convex programming contains both hard and easy solvable problems. If the feasibility problem in a convex program can be solved in polynomial times then these problems can be solved/approximated in polynomial time.

Every convex program can be formulated as cone program. Cone programming is well known from decades. Specifically its special cases like second order cone programming and/or Semidefinite programming are very well studied in literature. Feasibility problems for the semidefinite and second order cone programs can be solved in polynomial time hence these two classes are polynomial time solvable/approximable. Another subclass of cone programming is copositive programming which is rather new and can be applied to many combinatorial optimization problems. Since the feasibility problem for copositive programming cannot be determined in polynomial time hence existence of a polynomial time algorithm for this class of problem is out of question unless $P = NP$.

The notion of semi-infinite programming is used to denote the class of optimization problems in which the number of variables are finite but constraints are infinite. This class contains both convex and nonconvex optimization problem. Cone programs can be formulated as a special sub class of semi-infinite programming as we will see in chapter 5.
1.2 Problem Statement

Semidefinite programming (SDP) is a very well studied problem so is semi-infinite programming (SIP). In contrast to SDP and SIP, copositive programming (CP) is a relatively new area of research. CP, SIP and SDP have some connections; we will investigate these connections in this thesis. Integer programming is an important class of mathematical programming. In terms of CP and SDP relaxations of hard combinatorial optimization problems, integer programming has some obvious connections with SDP and CP.

This thesis is mainly concerned with following questions,

1. Survey of copositive programming and its application for solving integer programs.
2. Semi-infinite representation of copositive and semidefinite programming

1.3 Literature Review

The topic of the thesis is quite wide, covering large subclasses of mathematical programming. For ease of reading and presentation we will divide our literature review into different subsections. Although copositive programming and semidefinite programming are special classes of cone programming, we will discuss them in different subsections.

1.3.1 Integer Programming

The area of integer programming is as old as linear programming. The development of this theory is progressed with the progress of discrete optimization. There is an overlap of literature available on integer programming, semidefinite programming and copositive programming. Since the strength of copositive programming and semidefinite programming lies in relaxation of hard combinatorial optimization problems. These problems are first formulated as integer programs then relaxations are formulated.

Here we would like to mention some classics on the theory and application of integer programming. A classic is the book by Alexander Schrijver [129], this book covers both theoretical and practical aspects of integer programming. Another title covering history of integer programming is edited by Spielberg and Guignard-Spielberg [133]. The proceedings
of the conferences "Integer programming and combinatorial optimization" \cite{1,11,15,16,39,40,41,56,86,98} covers recent developments in the area of integer programming for combinatorial applications.

### 1.3.2 Cone Programming

Conic programming represents an important class of mathematical programming. Conic programming includes both linear and nonlinear programs. If the cone under consideration is convex, then we will have convex conic programs. Convex conic programming has a number of applications in engineering and economics. Let $K$ be any convex cone (see Definition \ref{definition:convex_cone}), and $K^*$ is its dual then we consider the following optimization problem,

$$
\min \quad \langle C, X \rangle \\
\text{s.t.} \\
\quad \langle A_i, X \rangle = b_i, \quad \forall i = 1, \ldots, m \\
\quad X \in K
$$

where $A_i, C \subset S_n$ and $K \subset S_n$ is a convex cone and $\langle C, X \rangle = \text{trace}(C^TX)$ is the standard inner product. The dual of the above program can be written as follows

$$
\max \quad b^Ty \\
\text{s.t.} \\
\quad \sum_{i=1}^{m} y_i A_i + Z = C \\
\quad y \in \mathbb{R}^m, \; Z \in K^*
$$

Cone programming can be seen as general abstraction of linear programming. Semidefinite programming and copositive programming are two well known subclasses of cone programming. Most of the theoretical results and algorithms for Linear programming (LP) can be generalized in a straightforward manner to the case of cone programming. But there are some differences. Existence of a feasible solution in linear programming results in a zero duality gap for the primal and dual program. This is not true in general for cone programming see \cite{111}.
1.3. LITERATURE REVIEW

1.3.3 Semidefinite Programming

Perhaps the most well studied special class of conic programming is semidefinite programming (SDP). SDP are optimization problems over the cone of semidefinite matrices. SDP are natural extension of LP, where linear (in)equalities are replaced by semidefiniteness conditions. The earliest discussion on theoretical aspects of SDP was done in 1963 (for details see [138]). After initial discussions on theoretical aspects, a lot of researchers have paid attention to this subject. Now the theory of SDP has become very rich and most aspects like duality, geometry etc are very well discussed in literature. The well known book edited by Wolkowicz et al [143] contains nice articles on both theoretical aspects of SDP and its applications. Interior point methods for linear programming were introduced in 1984. In 1988, Nesterov and Nemirovsky [138] proved that by adaption of suitable barrier functions interior point methods can be defined for a general class of convex programming. Independent from Nesterov and Nemirovsky, Alizadeh [4] has specialized interior point methods for SDP. By adaption of suitable barrier function one can solve/approximate SDP in polynomial time with the help of interior point methods.

Semidefinite programs are well known for their applications in combinatorial optimization. The use of SDP for combinatorial optimization was known from 1973 (see [138]), but a first remarkable result in this area was obtained by Lovasz [102], who used semidefinite programs to bound the stability number of a graph by the so called theta number. In 1987, Shor (see Shor [131] or Vandenbeghe and Boyed [138]) gave his so called Shor relaxation for general quadratic optimization problems with quadratic constraints. Shor’s relaxation and the newly developed interior point methods for SDP have revolutionized the study of SDP and its application in combinatorial optimization. The real break through in SDP was achieved by Goemans and Williamson [68], when they found a randomized approximation algorithm for the Max Cut problem. After the Max-cut results this field got attention of many researchers and number of nice results have been obtained (see [5, 46, 66, 75, 94, 96, 123, 138, 143]). Wolkowicz [142] has collected a huge list of references with comments on different aspects of cone and semidefinite programming.

1.3.4 Copositive Programming

Another interesting class of cone programming is copositive programming (CP). That is cone programs over the cone of copositive matrices. CP is not solvable in polynomial time. The main difficulty arises for checking the membership of a matrix in the cone of
copositive or completely positive matrices. There is no efficient algorithm known for the solution of copositive programs. There exists some approximation methods for solving copositive programs. Beside the difficulty in the solution of copositive programs, a number of combinatorial optimization problems have been modeled as copositive programs [30, 52, 115].

The study of copositive and completely positive matrices started with the work on quadratic forms. An earliest definition of copositive matrices can be found in the work of Motzkin [107]. Hall [70], has introduced the notion of completely positive matrices and provided a first example that a "doubly nonnegative" matrix may not be completely positive. Other classic contributions for copositive and completely positive matrices are given by [73, 105]. The most recent works covering theoretical and practical aspects of copositive and completely positive matrices includes a book by Berman and Shaked-Monderer [12], a thesis by Bundfuss [30] and an unpublished manuscript [80]. Moreover the interested reader may also find article [6, 22, 31, 36, 42, 48, 53, 81, 126, 87, 137] interesting with respect to theoretical properties and characterizations of completely and copositive matrices. Although the list is not complete in any sense, it covers some classic articles dealing with some state of the art results on copositive and completely positive matrices.

The relation of copositivity and optimization was known as early as 1989 [19]. Moreover Danninger [42] has discussed the role of copositivity in optimality criteria of nonconvex optimization problem. The use of copositive programming relaxation for hard combinatorial optimization problem was started by the paper of Preisig [118]. In 1998, Quist et al [119] gave a copositive programming relaxation for general quadratic programming problem. Recently after Quist et al, Bomze and De Klerk [25] applied the copositive programming to standard quadratic optimization problems and gave an approximation algorithm for the solution. To our knowledge the following is a complete list of all problem where copositive programming is applied: standard quadratic optimization problem [24], stable set problem [45, 51, 113], quadratic assignment problem [117], graph tri-partitioning [116], graph coloring [51, 69], clique number [17] and crossing number of graph [44]. Burer [34], has given a relaxation of a special class of general quadratic problems and proved with the help of some redundant constraint that his relaxation is exact.
1.3.5 Semi-infinite Programming

Semi-infinite programming is perhaps one of the oldest branches of mathematical programming. A semi-infinite program is an optimization problem of the form,

$$\min_x f(x)$$

$$s.t. \quad g_i(x) \leq 0 \quad \forall \ i \in V$$

where $f, g$ are real valued functions while $V$ is any compact set. As one can see, semi-infinite programs are mathematical programs over finite variables with infinite constraints, so is the name semi-infinite. It is a well known fact that some cone programs can be expressed as semi-infinite programs, see [90]. Semi-infinite programming has many application in engineering and science [101]. SDP can be converted to a semi-infinite programming problem [90]. Duality theory for semi-infinite programming is quite rich, a number of surveys and books are available discussing theoretical and practical aspects of semi-infinite programming [62, 63, 64, 101, 122, 130]. Lopez and Still have collected a huge list of literature available for semi-infinite programming [100]. There does not exist an algorithm which can solve all kind of semi-infinite optimization problem. In fact this is still an active area of research in semi-infinite programming to find a good algorithm for solving a class of semi-infinite programming problems.

1.4 Structure of the Thesis

This thesis consists of totally five chapters. Chapter one (the present chapter) deals with the introduction and literature review of the thesis topic.

In the second chapter we will briefly discuss the cones of matrices and their properties. We will also discuss copositive and completely positive matrices in detail in chapter 2. Moreover this chapter also contains notations we will use throughout this thesis.

Chapter 3 will contain an introduction of cone programming. We will also discuss semidefinite programming and its relation with quadratic programming.

Chapter 4 will deal with copositive programming. We will overview the combinatorial optimization problems where copositive programming is applied. We will also briefly describe the algorithm available for solution/approximation of copositive programs.

Chapter 5 is the final chapter and deal with an introduction to semi-infinite program-
1.4. STRUCTURE OF THE THESIS

ming. We will also state and prove strong duality result for Linear Semi-infinite programs. Later we will use this strong duality result to establish similar result for copositive and semidefinite programs. In the last section of this chapter we will discuss the semi-infinite programming approach for the solution of semidefinite programs.
Chapter 2

Cones and Matrices

The aim of this chapter is to introduce some basics on cones of matrices and related results which we will need in our next chapters. The first section of this chapter is concerned with some basic definition and self duality of semidefinite cones. The second section will deal with the copositive matrices and cones. The last section will deal with the completely positive cones. In the last section we will also show that the dual of the copositive cone is the completely positive cone and vice versa.

2.1 Basic Definitions

Definition 2.1 (Kronecker Product). Let $A \in \mathbb{R}^{(m \times n)}$ and $B \in \mathbb{R}^{(p \times q)}$ then the Kronecker product denoted by $\otimes$ is given by,

$$A \otimes B = \begin{bmatrix}
a_{11}B & \cdots & a_{1n}B \\
\vdots & \ddots & \vdots \\
a_{m1}B & \cdots & a_{mn}B
\end{bmatrix}$$

Definition 2.2 (Inner Product). The standard inner product denoted by $\langle . , . \rangle$ is given by $\langle X, Y \rangle = \text{trace}(X^T Y) = \sum_i \sum_j x_{ij}y_{ij}$ for $X, Y \in \mathbb{R}^{m \times n}$, where $\text{trace}(A = \{a_{ij}\}) = \sum_{i=1}^n a_{ii}$ is the trace of a matrix $A \in \mathbb{R}^{(n \times n)}$.

Definition 2.3 (Convex Set). A subset $S \subset \mathbb{R}^n$ is called convex if $\lambda x + (1 - \lambda)y \in S$ for all $x, y \in S$ and $0 \leq \lambda \leq 1$. The convex hull of $S$ denoted by $\text{conv}(S)$ is the minimal convex set which contains $S$. 
2.1. BASIC DEFINITIONS

Definition 2.4 (Extremal Points). A point $y$ of convex set $S$ is extremal point if it cannot be represented as convex combination of two points different from $y$. In other words a representation $y = y_1 + y_1, y_1, y_2 \in S$ is possible if and only if $y_1 = \lambda y, y_2 = (1 - \lambda)y$ where $0 \leq \lambda \leq 1$.

Definition 2.5 (Extreme Ray). A face of a convex set $S$ is a subset $S' \subseteq S$ such that every line segment in $S$ which has a relative interior point in $S'$ must have both end points in $S'$. An extreme ray is a face which is closed half-line.

Definition 2.6 (Convex Cone). A set $K \subset \mathbb{R}^{(m\times n)}$ which is closed under nonnegative multiplication and addition is called convex cone, i.e.,

$$X, Y \in K \Rightarrow \lambda (X + Y) \in K \; \forall \lambda \geq 0$$

A cone is pointed if $K \cap -K = \{0\}$. The dual of a cone $K$ is a closed convex cone denoted by $K^*$ and is given by,

$$K^* = \{Y \in \mathbb{R}^{m\times n} : \langle X, Y \rangle \geq 0, \; \forall X \in K\}$$

where $\langle ., . \rangle$ stands for the standard inner product.

Definition 2.7 (Recession Cone). Let $S \subset \mathbb{R}^n$ be a convex set then the recession cone is the set of vectors $d \in \mathbb{R}^n$ such that

$$rec(S) = \{d \in \mathbb{R}^n : x + \lambda d \in S, \; \forall x \in S, \; \lambda \geq 0\}$$

where $rec(S)$ denotes the recession cone of the set $S$.

2.1.1 Notation

Here we will enlist notations which we will use in next sections and chapters. Some of them are already mentioned before, we will repeat them.

We will use $\otimes$ to denote the standard Kronecker product, $vec(A)$ will denote the matrix $A \in \mathbb{R}^{m\times n}$ when written, column wise, as a vector in $\mathbb{R}^{mn}$ and $conv(S)$ will denote the convex hull of a set $S$. If $P$ denotes a program then $val(P)$ will denote the optimum value of the program and $Feas(P)$ will denote the set of feasible points of the program $P$. $N_n$, $S_n$, $S_n^+$, $C_n$, $C_n^*$ will denote the cone of nonnegative, symmetric, semidefinite, copositive and completely positive matrices respectively. We will use subscript such as $x_i$
2.2 Semidefinite Matrices and Cones

Semidefinite matrices are very well studied in literature. We will give the definition of semidefinite matrices and cones generated by such matrices. We will start by defining nonnegative matrices,

**Definition 2.8 (Nonnegative Matrix).** An \( m \times n \) matrix is nonnegative if all its entries are nonnegative. If \( A \) in nonnegative then we will write \( A \geq 0 \). The cone generated by all \( n \times n \) nonnegative matrices will be denoted by \( N_n \).

**Definition 2.9 (Symmetric Matrix).** An \( n \times n \) matrix \( A \) is called symmetric if \( A^T = A \), where \( A^T \) denotes the transpose of matrix \( A \). The cone of symmetric matrices denoted by \( S_n \) is the cone generated by all symmetric matrices.

Semidefinite matrices are very well studied due to their large application in system and control engineering and many other areas of science. Here we will only define semidefinite matrices and establish the self duality result of the semidefinite cone \( S_n^+ \).

**Definition 2.10 (Semidefinite Matrix).** An \( n \times n \) symmetric matrix \( A \) is called semidefinite if \( x^T Ax \geq 0 \), \( \forall x \in \mathbb{R}^n \).

The set of all \( n \times n \) semidefinite matrices define a cone called the cone of semidefinite matrices. We will denote this cone by \( S_n^+ \) and if \( A \in S_n^+ \) we will write \( A \succ 0 \).

Lemma 2.11 gives the duality result for semidefinite cone,

**Lemma 2.11.** The cone of semidefinite matrices is self dual, i.e., \( S_n^+ = (S_n^+)^* \)

**Proof.** \( S_n^+ \subseteq (S_n^+)^* \): It is not difficult to show that \( X, Y \in S_n^+ \), implies \( \langle X, Y \rangle \geq 0 \).

\((S_n^+)^* \subseteq S_n^+ \): Let \( X \in (S_n^+)^* \), then for all \( x \in \mathbb{R}^n \) the matrix \( xx^T \) is positive semidefinite. So,

\[ 0 \leq \langle X, xx^T \rangle = \text{trace}(X xx^T) = x^T X x \]

Hence \( X \in S_n^+ \).

This completes the proof. 

\[ \square \]
2.3

Copositive Matrices

Copositive matrices were introduced in 1952 and can be defined as follows,

**Definition 2.12 (Copositive Matrices).** A symmetric matrix $X$ of order $n$ is copositive if

$$x^T X x \geq 0 \quad \forall \ x \in \mathbb{R}_n^+$$

Alternatively, we can define copositive matrices as,

**Definition 2.13.** A symmetric matrix $A$ is copositive if and only if the optimization problem

$$\min_{\|x\|=1} \ x^T A x$$

has a nonnegative optimal value.

**Remark** Bundfuss and Dür [31] have described an algorithmic approach for testing copositivity of a matrix based on the above definition. We will briefly discuss their approach in section 4.3.3. For complete details see [31].

The set of all symmetric matrices of order $n$ generate a cone called cone of copositive matrices and is given by,

$$C_n = \{ X \in S_n : v^T X v \geq 0 \forall v \in \mathbb{R}_n^+ \}$$

We will write $A \in C_n$ to describe that $A$ is copositive or $A$ is in cone of copositive matrices.

Checking copositivity is hard. For matrices of order less than 4, there exists some characterization based on the structure of matrices. These condition can be checked easily, for details see [80]. Another interesting result states that matrices of order less than 4 are copositive if and only if they can be decomposed as the sum a nonnegative and a semidefinite matrix, see Theorem 2.14.

**Theorem 2.14.** Let $n \leq 4$ then $C_n = S_n^+ + N_n$.

**Proof.** Consider $S_n^+ + N_n \subset C_n$. Since both $S_n^+ \subset C_n$ and $N_n \subset C_n$ and $C_n$ is a cone hence $S_n^+ + N_n \subset C_n$.

The other part of the proof is quite complicated and lengthy. We refer the interested reader to [48].
2.3. COPOSITIVE MATRICES

For $n > 4$ we will have strict inclusion $S_n^+ + N_n \subset C_n$. The following counter example shows that strict inclusion holds for $n > 4$.

Example Consider the so called Horn-matrix

$$A = \begin{bmatrix}
1 & -1 & 1 & 1 & -1 \\
-1 & 1 & -1 & 1 & 1 \\
1 & -1 & 1 & -1 & 1 \\
1 & 1 & -1 & 1 & -1 \\
-1 & 1 & 1 & -1 & 1 \\
\end{bmatrix}$$

Let $x \in \mathbb{R}^n$, then,

$$x^T Ax = (x_1 - x_2 + x_3 + x_4 - x_5)^2 + 4x_2x_4 + 4x_3(x_5 - x_4)$$

$$= (x_1 - x_2 + x_3 - x_4 + x_5)^2 + 4x_2x_5 + 4x_1(x_4 - x_5)$$

If $x$ is nonnegative and $x_5 \geq x_4$ then $x^T Ax \geq 0$ for the first expression. For the second expression above we should have $x_5 < x_4$ to obtain $x^T Ax \geq 0$ for all nonnegative $x$. Hence $A$ is copositive and $A \notin S_n^+ + N_n$ (for complete details see [73]).

There are many characterizations of copositive matrices. But none of them is practical except for some special cases. For tridiagonal and acyclic matrices, copositivity can be tested in polynomial time (see [21, 82]). The copositivity of a matrix has relations with the copositivity of principle sub-matrices. This fact is given in Proposition 2.15.

Proposition 2.15. If $A$ is copositive then each principle sub-matrix of order $n - 1$ is also copositive.

The converse of the above proposition is not true in general. However, in particular cases the converse is also true,

Theorem 2.16. Let $A \in S_n$ and each principle sub-matrix of order $n - 1$ is copositive. Then $A$ is not copositive if and only if $A^{-1}$ exists and is entry wise non-positive.

The matrix of the form

$$\begin{pmatrix} 1 & x \\ x & X \end{pmatrix}$$

where $x \in \mathbb{R}^n, X \in \mathbb{R}^{n \times n}$ is often used in combinatorial application of copositive programming. Theorem 2.17 states the criteria for testing copositivity of this kind of matrices.
### 2.4. COMPLETELY POSITIVE MATRICES

**Theorem 2.17.** Let \( x \in \mathbb{R}^n \) and \( X \in S_n \). The matrix

\[
A = \begin{pmatrix}
a & x \\
x & X
\end{pmatrix} \in S_{n+1}
\]

is copositive if and only if

1. \( a \geq 0, X \) copositive
2. \( y^T(aX - xx^T)y \geq 0 \) for all \( y \in \mathbb{R}_+^n \) such that \( x^Ty \leq 0 \)

### 2.4 Completely Positive Matrices

One can define the completely positive matrices as follows,

**Definition 2.18 (Completely Positive Matrix).** A symmetric \( n \times n \) matrix \( A \) is called completely positive if it can be factorized such that \( A = BB^T \) where \( B \geq 0 \) is an arbitrary nonnegative \( n \times m \) matrix.

The set of all completely positive matrices generate a cone called the cone of completely positive matrices, which can be defined as,

\[
CPP_n = \left\{ X \in S_n : X = \sum_{k=1}^{N} y^k(y^k)^T \text{ with } \{y^k\}_{k=1}^{N} \subseteq \mathbb{R}_+^n/\{0\}, N \in \mathbb{N} \right\} \cup \{0\}
\]

It is interesting to note that copositive and completely positive matrices are dual to each other. In Lemma 2.19 we will proof this fact,

**Lemma 2.19.**

The dual of \( C_n \) is \( CPP_n \) and vice versa i.e.

\[
C_n = (CPP_n)^* \quad \text{and} \quad C_n^* = CPP_n
\]

**Proof.** First We will show,

\[
C_n = (CPP_n)^*) \quad \text{ (2.1)}
\]

\( C_n \subset (CPP_n)^* \): Let \( X \in C_n \) and \( Y \in CPP_n \). Since \( Y \in CPP_n \) so there exists finitely many
2.4. COMPLETELY POSITIVE MATRICES

vectors \( y_i \in \mathbb{R}_+^n / \{0\}, i \in N \subset \mathbb{N} \) such that \( Y = \sum_i y_i y_i^T \). Now we consider,

\[
\langle X, Y \rangle = \left\langle X, \sum_i y_i y_i^T \right\rangle = \sum_i y_i^T X y_i \geq 0
\]

\((CPP_n)^* \subset C_n : \) Consider \( A \notin C_n \). Then there exists \( x \geq 0 \) such that \( x^T x, A = x^T A x < 0 \). Since \( xx^T \in CPP_n \) (by definition of \( CPP_n \)) and \( \langle x^T x, A \rangle < 0 \) so \( A \notin (CPP_n)^* \).

Hence \((CPP_n)^* \subset C_n\)

For \( C_n^* = CPP_n \), we will consider \([2.1]\) and take dual on both sides to get

\[
C_n^* = (CPP_n^*)^* = CPP_n
\]

The last equality follows from well known results, "if a cone \( K \) is closed and convex then \((K^*)^* = K\)."

Since \( C_n^* = CPP_n \) in the rest of thesis we will use \( C_n^* \) to represent cone of completely positive matrices.

The cone of completely positive matrices is interesting with respect to combinatorial application of copositive programming. Unlike copositive matrices the cone of completely positive matrices is contained in the cone of semidefinite matrices hence we have,

\[
C_n^* \subset S_n^+ \subset C_n
\]

Just like copositive matrices for matrices of order less than four it is easy to check their membership in the cone of completely positive matrices,

**Proposition 2.20.** Let \( n \leq 4 \) then \( C_n^* = S_n^+ \cap N_n \).

**Proof.** The inclusion \( C_n^* \subset S_n^+ \cap N_n \) follows from definition of completely positive matrices. Since \( A \in C_n^* \) if and only if \( A = BB^T \) where \( B \in N_n \) hence \( A \in N_n \). Moreover \( A \in S_n^+ \) since \( x^T A x = x^T \sum_{k=1}^N (y^k)^T y^k x = \sum_{k=1}^N (y^k x)^T (y^k x) \geq 0 \). For the other side of the inclusion see [105].

In the literature matrices of the form \( S_n^+ \cap N_n \) are called doubly nonnegative matrices. For arbitrary \( n \) the inclusion \( C_n^* \subset S_n^+ \cap N_n \) is always true. But the other side of inclusion is not true in general,
2.4. COMPLETELY POSITIVE MATRICES

Example

\[ A = \begin{bmatrix}
1 & \frac{1}{2} & 0 & 0 & \frac{1}{2} \\
\frac{1}{2} & 1 & \frac{1}{2} & 0 & 0 \\
0 & \frac{1}{2} & 1 & \frac{3}{4} & 0 \\
0 & 0 & \frac{3}{4} & 1 & \frac{1}{2} \\
\frac{1}{2} & 0 & 0 & \frac{1}{2} & 1
\end{bmatrix} \]

It is clear that \( A \in N_n \), also \( A \in S_n^+ \) since,

\[
x^T Ax = \left( \frac{1}{2} x_1 + x_2 + \frac{1}{2} x_3 \right)^2 + \left( \frac{1}{2} x_1 + \frac{1}{2} x_4 + x_5 \right)^2
\]

\[ + \frac{1}{2} \left( x_1 - \frac{1}{2} x_3 - \frac{1}{2} x_4 \right)^2 + \frac{5}{8} (x_3 + x_4)^2 \]

but \( A \) is not completely positive (for a detailed proof see [72, page 349]).

Remark For a matrix of order less than five it is easy to check if the matrix is completely positive. There exists examples where matrices of order five are doubly nonnegative but not completely positive, hence doubly nonnegative matrices of order five got special attention of researchers (see [12, 13, 36, 99, 145] and references there in).

By the definition, one can see that checking if \( A \) is completely positive amounts to checking if there exists a matrix \( B \in \mathbb{R}^{n \times m}_+ \) such that \( A = BB^T \). It is not trivial to find this kind of factorization. A big part of literature on completely positive matrices deals with finding the least number \( m \) for which this factorization is possible. The minimal number \( m \), the so called cp-rank, is conjectured to be equal to \( \left\lfloor \frac{n^2}{4} \right\rfloor \) where \( n \) is the order of the matrix.

As stated earlier the matrix of the form \( \begin{bmatrix} 1 & x \\ x & X \end{bmatrix} \) often occurs in combinatorial application of copositive programming. One natural question arises if we are given with \( X \in C_n^* \), can we construct a factorization of the matrix \( \begin{bmatrix} 1 & x \\ x & X \end{bmatrix} \). Recently Bomze [22], has tried to answer this question by giving sufficient conditions under which we can determine the complete positivity of a matrix given the complete positivity of a principle block.

Another important property which is often used in SDP relaxation of hard combinatorial problem is: \( \begin{bmatrix} 1 & x \\ x & X \end{bmatrix} \in S_n^+ \) if and only if \( X - xx^T \in S_n^+ \). A natural question arises, can we generalize this result to the case of completely positive matrices. In order to answer this question we will start with following Lemma [2.21]
Lemma 2.21. Let \( x \in \mathbb{R}^n_+ \) and \( X - xx^T \in C_n^* \) then,

\[
A = \begin{pmatrix} 1 & x \\ x & X \end{pmatrix} \in C_n^*
\]

Proof. Let

\[
Y = \begin{pmatrix} a & \hat{x} \\ \hat{x} & \hat{X} \end{pmatrix} \in C_{n+1}
\]

where \( a \geq 0, \hat{X} \in C_n \) and consider,

\[
\langle A, Y \rangle = a + 2\hat{x}^T x + \langle X, \hat{X} \rangle
\]

\[
= a + 2\hat{x}^T x + \langle X - xx^T, \hat{X} \rangle + \langle xx^T, \hat{X} \rangle
\]

Since \( \langle X - xx^T, \hat{X} \rangle \geq 0, \langle xx^T, \hat{X} \rangle \geq 0 \), by duality of copositive and completely positive matrices. Hence nonnegativeness of the above expression depends on \( \hat{x}^T x \), so we will consider two cases,

**Case 1:** \( \hat{x}^T x \geq 0 \)

\[
\langle A, Y \rangle = a + 2\hat{x}^T x + \underbrace{\langle X - xx^T, \hat{X} \rangle}_{\geq 0} + \underbrace{\langle xx^T, \hat{X} \rangle}_{\geq 0}
\]

\[
\geq 0
\]

**Case 2:** \( \hat{x}^T x \leq 0 \ a > 0 \):

\[
\langle A, Y \rangle = a + 2\hat{x}^T x + \langle X - xx^T, \hat{X} \rangle + \langle xx^T, \hat{X} \rangle
\]

\[
= \frac{1}{a} \left( a + (\hat{x}^T x)^T (\hat{x}^T x) \right)^2
\]

\[
+ x^T(\hat{X} - \hat{x}^T \hat{x}) x + \langle X - xx^T, \hat{X} \rangle
\]

By Theorem 2.17 we have, if \( \hat{x}^T x \leq 0 \) then \( x^T(a\hat{X} - \hat{x}\hat{x}^T)x \geq 0 \) or \( x^T \left( \hat{X} - \frac{1}{a}\hat{x}\hat{x}^T \right) x \geq \)
0. Hence we will have,

\[
\langle A, Y \rangle = \frac{1}{a} \left( a + (\hat{x}^T x) \left( \hat{x}^T x \right) \right)^2 \\
+ x^T \left( \hat{X} - \frac{1}{a} \hat{x} \hat{x}^T \right) x \\
\geq 0 \\
+ \left( \langle X - xx^T, \hat{X} \rangle \right) \\
\geq 0
\]

\[
a = 0: \text{Since } \hat{x}^T x \leq 0 \text{ so by theorem } 2.17
\]

\[
(x^T \left( a \hat{X} - \hat{x} \hat{x}^T \right) x) = -(\hat{x}^T x)^2 \geq 0
\]

\[
giving that \hat{x}^T x = 0.
\]

So for an arbitrary copositive matrix \( Y \), \( \langle A, Y \rangle \) is nonnegative hence \( A \) is completely positive. This completes the proof. \( \square \)

The converse of Theorem 2.21 namely \( (\begin{array}{c} x \\ X \end{array}) \in C_n^* \) implies \( X - xx^T \in C_n^* \) is not true in general. The following is a counter example.

**Example** Let,

\[
A = \begin{pmatrix}
1 & 2 & 1 \\
2 & 5 & 1 \\
1 & 1 & 3
\end{pmatrix}
\]

It is clear that \( A \in N_3 \) and \( A \in S_3^+ \) since,

\[
x^T Ax = 3 \left( x_3 + \frac{1}{3} (x_1 + x_2) \right) + \frac{14}{3} \left( x_3 + \frac{5}{14} \right)^2 + \frac{1}{14} x_1^2 \geq 0 \quad \forall \ x \in \mathbb{R}^3
\]

Hence by Proposition 2.20, \( A \) is completely positive. Now take,

\[
X = \begin{pmatrix}
5 \\
1 \\
3
\end{pmatrix}, \ x = \begin{pmatrix}
2 \\
1
\end{pmatrix}
\]
Then we will have,

\[ X - xx^T = \begin{pmatrix} 1 & -1 \\ -1 & 2 \end{pmatrix} \]

Since \( X - xx^T \), has negative entries, so it cannot be completely positive.

**Remark** Let \( A = \begin{pmatrix} 1 & x \\ x & X \end{pmatrix} \) and \( x \in \{0, 1\}^n, X \in \{0, 1\}^{n \times n} \) then the converse of Lemma 2.21 is also true owing to the result "A symmetric binary matrix is semidefinite if and only if it is completely positive [97, Corollary 1]."

The interior of a cone plays an important role for establishing strong duality results. The characterization of the interior is often required for checking strong duality. The interior of copositive matrices consists of the so called strictly copositive matrices (\( A \in S_n \) is strictly copositive if and only if \( x^T A x > 0, \quad \forall x \in \mathbb{R}_n^+ \{0\} \)). For the case of completely positive matrices, characterization of the interior is not simple. Dür and Still [53] have given a characterization for the interior of completely positive matrices,

**Theorem 2.22.** Let \([A_1|A_2]\) describe the matrix whose columns are columns of \( A_1 \) augmented with columns of \( A_2 \) then we can define the interior of the completely positive matrix as,

\[ \text{int}(C_n^*) = \{ AA^T | A = [A_1|A_2] \text{where } A_1 > 0 \text{ is nonsingular and } A_2 \geq 0 \} \]

**Proof.** For a proof see [53]. \(\square\)
Chapter 3

Cone and Semidefinite Programming

Conic programs are similar to linear programs in a sense that we have a linear objective function with linear constraints but there is an additional constraint which checks the membership of the variable in a cone. The well known examples of conic programs are linear programming, semidefinite programming and copositive programming. These three examples are respectively programs on the nonnegative cone, semidefinite cone and copositive cone.

In this chapter we will study the primal dual formulation of general conic programs and then briefly discuss the duality theory of conic programs. The last sections of this chapter are concerned with Semidefinite programming (SDP) and SDP relaxation of quadratic programming. We will also describe some application of SDP.

3.1 Cone Programming

A large class of mathematical programs can be represented as a conic program. We will consider the following formulation of a conic program

$$\min_{X} \langle C, X \rangle$$

s.t.

$$Cone_P \quad \langle A_i, X \rangle = b_i, \quad \forall i = 1, \ldots, m$$

$$X \in K$$
3.1. CONE PROGRAMMING

where $A_i, C \in S_n$ and $K$ is a cone of symmetric $n \times n$ matrices. The dual of the above
program can be written as follows

$$\max_y \ b^T y$$

$$\text{s.t.}$$

$$\text{Cone}_D \quad \sum_{i=1}^{m} y_i A_i + Z = C$$

$$y \in \mathbb{R}^m, Z \in K^*$$

3.1.1 Duality

In mathematical programming duality theory plays a crucial role for finding optimality
conditions and solution algorithms. Duality theory can be further classified into two cat-
egories weak duality and strong duality. In weak duality we investigate, if the optimal
value of the primal problem is bounded by the dual problem. Strong duality investigates
the conditions under which strict equality holds in the values of primal and dual solutions.
$\text{Cone}_P$ and $\text{Cone}_D$ satisfy weak duality. This fact is proved in Lemma 3.1.

**Lemma 3.1 (Weak Duality).** Let $X$ and $(y, Z)$ be feasible solutions for $\text{Cone}_P$ and $\text{Cone}_D$
respectively then $b^T y \leq \langle C, X \rangle$

**Proof.** We have

$$b^T y = \sum_{i=1}^{m} b_i y_i = \sum_{i=1}^{m} y_i \langle A_i, X \rangle = \sum_{i=1}^{m} \langle y_i A_i, X \rangle = \langle \sum_{i=1}^{m} y_i A_i, X \rangle$$

$$= \langle C - Z, X \rangle = \langle C, X \rangle - \langle Z, X \rangle$$

$$\leq \langle C, X \rangle$$

\[ \square \]

In the case of $K$ being the nonnegative orthant whenever $\text{Cone}_P$ or $\text{Cone}_D$ are feasible we
have equality in the optimal values. If both $\text{Cone}_P$ and $\text{Cone}_D$ are feasible then we have
zero duality gap and both optimal values can be attained. Unfortunately this nice property
does not hold for more general classes of conic programs such as semidefinite programs, as
shown by the following example,
3.1. CONE PROGRAMMING

Example Consider the conic program with,

\[ C = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 \end{pmatrix}, A_1 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \]

\[ A_2 = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 2 \end{pmatrix}, b = \begin{pmatrix} 0 \\ 2 \end{pmatrix} \text{ and } K = K^* = S_n^+ \]

Then it is not difficult to verify that \( \text{val}(Cone_P) = 1 \) and \( \text{val}(Cone_D) = 0 \) even though both problems are feasible.

For strong duality in conic programs we need extra conditions on the constraints.

**Definition 3.2 (Primal Slater Condition).** We say a set of feasible points satisfies the Slater condition for \( Cone_P \) if there exists \( X \in \text{int}(K) \) such that \( \langle A_i, X \rangle = b_i \).

The Slater condition for dual problems can be defined in a similar manner. By taking this additional assumption we can derive a strong duality result as given in Theorem 3.3

**Theorem 3.3 (Strong Duality).** If the primal problem \( Cone_P \) or its dual \( Cone_D \) satisfies the Slater condition then, \( \text{val}(Cone_P) = \text{val}(Cone_D) \).

**Remark** The discussion of conic duality can be traced as early as 1958. This subject is more flourished in the early 1990’s. The backbone for the development of this area is the desire to extend Karmarkar’s polynomial time LP algorithm to the non-polyhedral case, see [111] and references there in.

3.1.2 Examples of Cone Programming

For different values of \( K \) and \( K^* \) we obtain a class of well known programming problems,

1. If \( K = \Re_n^+ \) the so called nonnegative orthant, we will have linear programming. The feasibility of one of the primal or dual program implies that strong duality holds. Linear programming has huge applications in science and engineering. In fact the term mathematical programming was first used as linear programming coined by Dantzig in 1940. The simplex method was considered to be the first method for solution of linear programs but now linear time algorithm like the ellipsoidal method and interior point method exists for linear programming problems (see [125, 141]).
2. If we consider \( K = K^* = \{ (\xi, x) \in \mathbb{R} \times \mathbb{R}^n | \xi \geq ||x||_2 \} \) the so called second order cone, then we will get the second order cone programming problem. In this case we need a Slater condition for obtaining strong duality. Polynomial time algorithms also exists for this class of problems. They have large application in control engineering.

3. For \( K = S_n^+ \) we obtain the semidefinite programming problems. Semidefinite programming has large application in combinatorial optimization, control engineering, eigenvalue optimization and many other areas of engineering and science. We will discuss this class of problems in more details in the next section.

4. For \( K = C_n \) we will obtain copositive programs. It is interesting to note that this is the first problem where the cone is not self dual. Moreover, no polynomial time algorithm exists for this class of problems. This is relatively new area of cone programming.

From the above examples it is evident that cone programming covers many mathematical programming problems. For some specific kind of cone it is easy to solve the cone programs but for other cases, we cannot solve them to optimality. Interior point methods are a common choice for solving conic program. But for certain classes of conic programs, like copositive programs, they failed to provide solutions \([30, 115]\).

We will close this section by arguing that any convex program can be written as cone programming problem. Consider the problem,

\[
\min_x c^T x \\
s.t. \\
\quad x \in Y
\]

where \( Y \) is any closed convex set in \( \mathbb{R}^n \). This problem can be seen as a conic programming problem, where the dimension of the problem is increased by one,

\[
\min_{x, \xi} c^T x \\
s.t. \\
\quad \xi = 1 \\
\quad (x, \xi) \in K
\]

with \( K = cl\{(x, \xi) \in \mathbb{R}^n \times \mathbb{R} | \xi > 0, x/\xi \in Y\} \) where \( cl(S) \) denotes the closure of the set \( S \).
Proposition 3.4. $K$ is closed convex cone.

Proof. It is not difficult to show that, if $Y \in K$ then $\lambda Y \in K$. Since $(y, \xi) \in K$ implies $\frac{y}{\xi} \in Y$, then for any $\lambda \geq 0 \frac{\lambda y}{\lambda \xi} \in Y$ giving $\lambda (y, \xi) \in K$.

It is well known that if a set is convex, then its closure is also convex and vice versa. So let us consider the set,

$$\hat{K} = \{(x, \xi) \in \mathbb{R}^n \times \mathbb{R} | \xi > 0, x/\xi \in Y\}$$

We will show that $\hat{K}$ is convex. Let $Y_1, Y_2 \in \hat{K}, 0 \leq \lambda \leq 1$ with $Y_i = (y_i, \xi_i) \in \mathbb{R}^n \times \mathbb{R}$, $\xi_i > 0$ and $y_i/\xi_i \in Y$ for $i = 1, 2$, where $Y$ is a convex set.

$$\frac{\lambda y_1 + (1 - \lambda)y_2}{\lambda \xi_1 + (1 - \lambda)\xi_2} = \frac{\lambda}{\lambda \xi_1 + (1 - \lambda)\xi_2} y_1 + \frac{(1 - \lambda)}{\lambda \xi_1 + (1 - \lambda)\xi_2} y_2$$

$$= \rho \frac{y_1}{\xi_1} + (1 - \rho) \frac{y_2}{\xi_2}$$

where

$$\rho = \frac{\lambda \xi_1}{\lambda \xi_1 + (1 - \lambda)\xi_2} \in [0, 1]$$

Since $Y$ is convex so, $\rho y_1/\xi_1 + (1 - \rho)y_2/\xi_2 \in Y$. Hence

$$\frac{\lambda y_1 + (1 - \lambda)y_2}{\lambda \xi_1 + (1 - \lambda)\xi_2} \in Y \Rightarrow (\lambda y_1 + (1 - \lambda)y_2, \lambda \xi_1 + (1 - \lambda)\xi_2) \in \hat{K},$$

So $\hat{K}$ is convex. Hence $K$ is a closed convex cone. 

3.2 Semidefinite Programming (SDP)

SDP can be regarded as one of the most well studied special cases of cone programming. As mentioned earlier SDP can be seen as natural generalization of linear programming where linear inequalities are replaced by semidefiniteness conditions. Moreover it is one of the polynomial time solvable(approximable) classes of cone programming. SDP has become a very attractive area of research among the optimization community due to its large applications in combinatorial optimization, system and control, solution of quadratically constrained quadratic programs, statistics, structural optimization and maximum eigenvalue problem.
3.2. SEMIDEFinite PROGRAMMING (SDP)

SDP are optimization problems often written in the following form,

\[
\min_X \langle C, X \rangle \\
\text{s.t.} \\
SDP \quad \langle A_i, X \rangle = b_i, \quad \forall \ i = 1, \ldots, m \\
X \in S_n^+ \ (\text{or } X \succeq 0)
\]

where \(A_i, C, X \in S_n\). The dual of the above program is given by,

\[
\max_y b^T y \\
\text{s.t.} \\
SDP_D \quad \sum_{i=1}^m y_i A_i + Z = C \\
y \in \mathbb{R}^m, Z \in S_n^+
\]

It is worth mentioning that the cone of semidefinite matrices is self dual, see Lemma 2.11. Being an extension of linear programming most of the algorithms available for linear programming can be generalized for SDP in a straightforward manner. Beside many similarities there are some major differences one of them is, as shown before, for general SDP strong duality does not hold in general. In linear programming a pair of complementary solutions always exists but this is not the case in SDP. For a rational linear program the solution is always rational but this not the case for SDP. A rational SDP may have an irrational solution. Moreover there does not exists a practical simplex like method for SDP (for details see [138]). As a special case of cone programming the same duality theory holds for SDP. Based on the structure of SDP many necessary and sufficient conditions are formulated for an optimal solution of SDP. For a detailed discussion on duality theory of SDP one can see [120, 121, 143]. Most appealing and useful application of SDP is SDP relaxation, which has a number of applications in combinatorial optimization.

Remark It is usual in combinatorial applications of SDP that strong duality holds, so optima can be obtained. SDP with zero duality gap were introduced by Borwein and Wolkowicz [28] and Ramana [120]. Ramana et al [121] has given a good comparison of zero duality gap results for SDP.
## 3.2. SEMIDEFINITE PROGRAMMING (SDP)

### 3.2.1 Integer Programming

Almost all combinatorial optimization problems can be formulated as integer programs. These programs may be linear or nonlinear depending on the specific problem. In this section we will deal with integer programs with only linear constraints and quadratic objective function. We consider the program,

\[
\min_x \; x^T Q x + 2c^T x \\
\text{s.t.} \\
\begin{align*}
IP & \quad a_i^T x \leq b_i, \quad \forall \; i = 1, \ldots, m \\
& \quad x_j \in \{0,1\} \quad \forall \; j = 1, \ldots, n
\end{align*}
\]

Most of the Combinatorial optimization problems can be expressed as IP. But unfortunately formulating a hard combinatorial problem as 0-1 integer program will not make the original problem tractable because 0-1 integer programming is itself a hard problem. It is well known that integer programming is NP-hard. The main difficulty lies in the integer constraints, i.e. the constraints which ensure that the solution must be integer. By relaxing this constraint one can obtain bounds on the solution of IP.

The first and the most common relaxation, the so called LP relaxation, is to replace the condition \( x \in \{0,1\} \) by the condition \( x \in [0,1] \). This kind of relaxation is useful for finding lower bounds on the optimum value of the original problem. We can state the linear programming relaxation of IP as follows,

\[
\min_x \; x^T Q x + 2c^T x \\
\text{s.t.} \\
\begin{align*}
IP_{LP} & \quad a_i^T x \leq b_i, \quad \forall i = 1, \ldots, m \\
& \quad x \in [0,1]^n
\end{align*}
\]

It is important to note that \( IP_{LP} \) is central to the branch and bound methods, used to approximate the solution of IP for details see \cite{57, 95}. In order to strengthen the relaxation \( IP_{LP} \) of IP one need to avoid fractional solution as much as possible. Chvatal-Gomory cuts (see \cite[Chapter 9]{55}, \cite{127, 129}) were introduced in order to avoid fractional solutions. Another important technique to strengthen \( IP_{LP} \) is lift and project methods developed in the nineties and proved very useful for obtaining good bounds on solution of IP.
3.2. **SEMIDEFINITE PROGRAMMING (SDP)**

[85, 94, 127, 129] for more references see [2].

Another common relaxation is the so called Lagrange relaxation which is based on the Lagrange function. The Lagrange relaxation results in another optimization problem and solving the dual problem will provide us with bounds on the optimum value, see [60, 96].

**SDP Relaxation**

Yet another relaxation coined by Lovasz [102], is to relax the integer constraint by the semidefinite constraint. This is possible due to the fact that $x_j \in \{0, 1\}$ can be equivalently written as $x_j^2 - x_j = 0$, hence $IP$ can be written in the form,

$$\min_x x^T Q x + 2c^T x$$

s.t.

$$a_i^T x \leq b_i, \quad \forall \ i = 1, \ldots, m$$
$$x_j^2 - x_j = 0 \quad \forall \ j = 1, \ldots, n$$

The SDP relaxation of the above program will be,

$$\min_x \langle X, Q \rangle + 2c^T x$$

s.t.

$$a_i^T x \leq b_i, \quad \forall \ i = 1, \ldots, m$$
$$X_{jj} = x_j$$
$$\begin{pmatrix} 1 & x \\ x & X \end{pmatrix} \succeq 0$$

The above relaxation is possible since $xx^T = X$ implies that there exists a rank one matrix $X = xx^T$. We can replace the rank one constraint with the semidefinite constraint obtaining the above relaxation. Another advantage is to gather all nonlinearities in one variable $X$, which is further relaxed with a semidefinitness condition.
3.2. SEMIDEFINITE PROGRAMMING (SDP)

3.2.2 Quadratically Constrained Quadratic Program

One can formulate the general quadratic programming problem with quadratic constraints as follows,

$$\begin{align*}
\min_x & \quad x^T Q x + 2(c^0)^T x \\
\text{s.t.} & \quad Q P \quad x^T A_i x + 2(c^i)^T x + b_i \leq 0 \quad \forall i \in \mathbb{I}
\end{align*}$$

where $Q, A_i, Q \in S_n$ and $\mathbb{I} = \{1, \ldots, m\}$.

It is worth mentioning that the above program may not be convex. A standard way to make this program convex is to gather all nonlinearities in one variable. For this we introduce a matrix $X$, such that $X = xx^T$ and consider, $x^T Q x = \text{trace}(Q xx^T) = \langle Q, xx^T \rangle = \langle Q, X \rangle$. Then the above program can be equivalently written as,

$$\begin{align*}
\min_X & \quad \langle Q, X \rangle + 2(c^0)^T x \\
\text{s.t.} & \quad Q P \quad \langle A_i, X \rangle + 2(c^i)^T x + b_i \leq 0, \quad \forall i \in \mathbb{I} \\
& \quad X = xx^T
\end{align*}$$

It is interesting to note that most of the conic relaxations in literature are based on introducing some new redundant constraints along with replacing the constraints $X = xx^T$ with the constraint of the form $X \in K$ where $K$ is a cone of matrices. If $K = S_n^+$ we will have semidefinite relaxation.

It is not difficult to verify that $x^T A_i x + 2(c^i)^T x + b_i = \left\langle \begin{pmatrix} b_i & (c^i)^T \\ c^i & A_i \end{pmatrix}, \begin{pmatrix} 1 & x^T \\ x & X \end{pmatrix} \right\rangle$, where $X = xx^T$. Let us define,

$$\mathcal{P} = \left\{ P_i = \begin{pmatrix} b_i & (c^i)^T \\ c^i & A_i \end{pmatrix} \quad \forall \ i \in \mathbb{I} \right\}$$
3.2. SEMIDEFINITE PROGRAMMING (SDP)

Then the set of feasible points of $QP$ will be defined as follows,

$$Feas(QP) = \left\{ x \in \mathbb{R}^n : \left\langle P_i, \begin{pmatrix} 1 & x^T \\ x & x^T \end{pmatrix} \right\rangle \leq 0, \ \forall \ P_i \in \mathcal{P} \right\}$$

The SDP relaxation of $QP$ relaxes the rank one matrix constraint $X = xx^T$ with the constraint $\begin{pmatrix} 1 & x^T \\ x & X \end{pmatrix} \succeq 0$, so we will have,

$$\min_{x, X} \left\langle Q, X \right\rangle + 2c^0^T x$$

s.t.

$$QP_{SDP} \quad \left\langle P_i, \begin{pmatrix} 1 & x^T \\ x & X \end{pmatrix} \right\rangle \leq 0, \ \forall \ i \in \mathbb{I}$$

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

Then the set of feasible points for the above program can be described by,

$$Feas(QP_{SDP}) = \left\{ x \in \mathbb{R}^n : \exists X \in S_n \text{ such that } \begin{pmatrix} 1 & x^T \\ x & X \end{pmatrix} \succeq 0 \right\} \quad \text{and} \quad \left\langle P_i, \begin{pmatrix} 1 & x^T \\ x & X \end{pmatrix} \right\rangle \leq 0, \ \forall \ i \in \mathbb{I} \right\}$$

Let us denote by $\mathcal{Q}_+$ all semidefinite quadratic functions i.e.

$$\mathcal{Q}_+ = \{ x^T A x + c^T x + b_i : A \in S_n^+, c \in \mathbb{R}^n, b_i \in \mathbb{R} \}$$

Let $\mathcal{K}$ denote the convex cone generated by the quadratic functions $x^T A_i x + 2(c_i)^T x + b_i$ i.e.

$$\mathcal{K} = cone \left\{ \mathcal{P} \right\}$$
Then we have following relaxation of QP

\[
\begin{align*}
\text{min}_x & \quad x^T Q x + 2(c^0)^T x \\
\text{s.t.} & \quad \left\langle P_i, \begin{pmatrix} 1 & x^T \\ x & xx^T \end{pmatrix} \right\rangle \leq 0 \quad \forall i \in I \\
& \quad P_i \in \mathcal{K} \cap \mathbb{Q}_+ 
\end{align*}
\]

The relaxations \( QP_{SDP} \) and \( QP_S \) are equal,

**Theorem 3.5.**

\[ \text{Feas}(QP_{SDP}) = \text{Feas}(QP_S) \]

**Proof.** One can define the dual cone of \( \mathcal{K} \) as,

\[ \mathcal{K}^* = \{ V \in S_{n+1} : \langle V, U \rangle \geq 0, \quad \forall U \in \mathcal{K} \} \]

We can see that for the set of feasible points for \( QP_S \) and \( QP_{SDP} \) we have,

\[
\begin{align*}
\text{Feas}(QP_{SDP}) &= \left\{ x \in \mathbb{R}^n : \exists X \in S_n \text{ such that } \begin{pmatrix} 1 & x^T \\ x & X \end{pmatrix} \in (\mathcal{K}^* \cap S_{1+n}^+) \right\} \\
\text{Feas}(QP_S) &= \left\{ x \in \mathbb{R}^n : \begin{pmatrix} 1 & x^T \\ x & xx^T \end{pmatrix} \in - (\mathcal{K} \cap \mathbb{Q}_+)^* \right\} \\
&= \left\{ x \in \mathbb{R}^n : \begin{pmatrix} 1 & x^T \\ x & xx^T \end{pmatrix} \in - \left( \mathcal{K}^* + \begin{pmatrix} 0 & 0^T \\ 0 & S_n^+ \end{pmatrix} \right) \right\}
\end{align*}
\]

\( \text{Feas}(QP_{SDP}) \subset \text{Feas}(QP_S) \): Take \( x \in \text{Feas}(QP_{SDP}) \), then there exists \( X \in S_n^+ \) such that \((x, X) \in \text{Feas}(QP_{SDP})\) then we can write

\[
\begin{pmatrix} 1 & x^T \\ x & xx^T \end{pmatrix} = \begin{pmatrix} 1 & x^T \\ x & X \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ 0 & xx^T - X \end{pmatrix}
\]

30
For $P_i \in \mathcal{P}$ we will have,

$$x^T A_i x + 2(c_i^T)^T x + b_i = \left\langle P_i, \begin{pmatrix} 1 \\ x \\ x \end{pmatrix} \right\rangle$$

$$= \left\langle P_i, \begin{pmatrix} 1 \\ x \\ x \end{pmatrix} \right\rangle + \left\langle A_i, \begin{pmatrix} x^T - X \\ \leq 0 \end{pmatrix} \right\rangle$$

$$\leq 0$$

So, $x \in \text{Feas}(QP_S)$.

**Feas**($QP_{SDP}$) $\supset$ Feas($QP_S$): For the converse let $x \in \text{Feas}(QP_S)$, then there exists some $H \in S^n_+$ such that,

$$\begin{pmatrix} 1 & x \\ x & xx^T + H \end{pmatrix} \in -K^*$$

The matrix $\begin{pmatrix} 1 & x \\ x & xx^T + H \end{pmatrix}$ is positive semidefinite if and only if $H$ is positive semidefinite. Which is our assumption that $H \in S^n_+$. Hence

$$\begin{pmatrix} 1 & x \\ x & xx^T + H \end{pmatrix} \in -K^* \cap S^n_+$$

So $x \in \text{Feas}(QSDP)$.

This completes the proof.

**Remark**
1. The above results even holds if the index set $I$ is infinite.
2. The relaxation $QP_S$ is first discussed by Fujie and Kojima [58] but its equality with the SDP relaxation is proved by Kojima and Tunçel [88].

**Another Semidefinite Relaxation**

Several semidefinite relaxations of $QP$ are discussed in the literature. The basic and simple semidefinite relaxation is given by Shor [131]. In the Shor relaxation we will relax
3.2. SEMIDEFINITE PROGRAMMING (SDP)

the constraint $X = xx^T$, by the constraint $X \in S^n_+$

$$\min_X \langle Q, X \rangle + 2c_0^T x$$

s.t.

$$\langle A_i, X \rangle + 2c_i^T x = b_i, \quad \forall i = 1, ..., m$$

$$X \in S^n_+$$

The Shor relaxation actually relax the rank one condition $X = xx^T$ by lifting matrix $X$ from the class of rank one matrices to the bigger class of semidefinite matrices.

The Shor-relaxation became central for finding the $\mu$-approximation algorithm for hard combinatorial optimization problems. Some nice $\mu$-approximation algorithms were found for problems like Max-Cut and box constrained problems (see [68, 119]).

3.2.3 Applications of SDP

Many state of the art results have been obtained by formulating the SDP relaxation of QP. We will enlist some of them,

1. The most popular SDP relaxation is for Max-Cut problem. Using a SDP relaxation along with randomization Goemans and Williamson[68] has obtained a 0.878-approximation algorithm for the Max-cut problem. This is the first major breakthrough for SDP. It has opened a way for application of SDP in combinatorial optimization problems. This problem is furthered discussed with reference to SDP by [124].

2. The SDP relaxation of stability number of a graph resulted in the so called Lovarz theta number. Theta number has not only provided a bound on the stability number of the graph but also provided a polynomial time algorithm for finding the stability number in perfect graph for details see [102, 113].

3. The well known spectral bundle methods for the eigenvalue optimization problem are based on the concept of SDP. These methods are used for finding maximum or minimum eigenvalues of a matrix. For details see [143].

4. SDP has been proved very useful for approximating nonlinear problems. Specifically
quadratically constrained quadratic programs (QCQP) are approximated by the use of SDP relaxations (for details see [9, 3, 138]).

5. There are many other complex problems for which SDP has provided promising results, this list of problems includes the satisfiability problem [7, 74], maximum clique and graph coloring [23, 49, 50], non-convex quadratic programs [58], graph partitioning [61, 115, 144, 146], nonlinear 0-1 programming [92, 97], knapsack problem [77], traveling salesman problem [47], quadratic assignment problem [115, 146], subgraph matching [128], statistics [143, Chapter 16 and 17] [138], control theory [138], structural design [138, 143, Chapter 15] and many other areas of science and engineering.
Chapter 4

Copositive Programming (CP)

Copositive programming is a relatively new area of research in cone programming. Only a few papers and a thesis have been written, covering applications and algorithms for copositive programming. The main theme of this chapter is to introduce copositive programming along with its application. We will also discuss some algorithms available for solving copositive programming. We will start with an introduction into copositive programming. Then we will give the applications of copositive programming. The final section of this chapter is concerned with the algorithm available for solving copositive programs.

4.1 Copositive Programming

Copositive programs refer to cone programs of the form,

\[
\begin{align*}
\min_x \quad & \langle C, X \rangle \\
\text{s.t.} \quad & \langle A_i, X \rangle = b_i, \quad \forall i = 1, ..., m \\
& X \in C_n
\end{align*}
\]
4.2. APPLICATIONS

where $A_i, C \subset S_n$, and the dual

$$\max_y b^T y$$

$$s.t.$$

$$CP_D \quad \sum_{i=1}^m y_i A_i + Z = C$$

$$y \in \mathbb{R}^m, Z \in C_n^*$$

with $C_n$ and $C_n^*$ are the cones of copositive and completely positive matrices respectively. It is not difficult to see that the above programs are convex. But being convex does not guaranty that we can solve these programs efficiently. The main difficulty lies in checking the set of feasible points of both the primal and dual programs. It is co-NP hard to check that a matrix is copositive (see Murty and Kabadi [110]).

Duality theory helps to find algorithms and solution procedures for mathematical programs. The same duality theory holds for copositive programs as for cone programs. As mentioned before interior of a cone plays an important role for developing strong duality results. Recently Dür and Still [53], has given characterization of the interior of the completely positive cone. In contrast to semidefinite programming duality theory for copositive programming is not very well discussed. A recent paper by Eichfelder and Jahn [54] has discussed the KKT type optimality condition along with duality for copositive programming.

4.2 Applications

Copositive programming has been applied to a number of interesting problems. Here we will briefly discuss these problems.
4.2. APPLICATIONS

4.2.1 Quadratic Programming

Consider the specific case of a quadratic program with only one quadratic constraint.

\[
\begin{align*}
\min_x & \quad x^T Q x \\
\text{s.t.} & \quad SQC \quad x^T A x = b \\
& \quad x \geq 0
\end{align*}
\]

where \( A \) is a strictly copositive matrix and \( b \in \mathbb{R} \).

The connection between copositive programming and the above problem was first investigated by Preisig [118]. \( SQC \) contains some interesting problems like standard quadratic programming over simplex and the maximum clique problem as special case. The copositive programming formulation of the above problem is,

\[
\begin{align*}
\min_X & \quad \langle Q, X \rangle \\
\text{s.t.} & \quad SQC_{C_n^*} \quad \langle A, X \rangle = b \\
& \quad X \in C_n^*
\end{align*}
\]

The dual formulation of the above program is,

\[
\begin{align*}
\max_y & \quad by \\
\text{s.t.} & \quad SQC_{C_n} \quad Q - Ay \in C_n
\end{align*}
\]

The set of feasible points of \( SQC_{C_n^*} \) is the intersection of the completely positive cone and a hyperplane. The extremal rays (see Definition 2.5) of the completely positive cone are rank one matrices \( xx^T \) with \( x \geq 0 \). So we have equivalence for \( SQC_{C_n^*} \) and \( SQC \) in the special case if we choose the hyperplane in such a way so that the extremal points (see Definition 2.4) of the feasible sets coincides with the extremal rays of completely positive cone. The equivalence of \( SQC \) and \( SQC_{C_n^*} \) is given in Lemma 4.1.
4.2. APPLICATIONS

Lemma 4.1. Let $A \in S_n$ be strictly copositive then

1. if $b > 0$ then the extremal points of the feasible set of $SQC_{C_n^*}$ are exactly the rank-one matrices $X = xx^T$ with $x^T Ax = b$ and $x \geq 0$.

2. The problems $SQC, SQC_{C_n^*}$ and $SQC_{C_n}$ are equivalent that is,

$$\text{val}(SQC) = \text{val}(SQC_{C_n^*}) = \text{val}(SQC_{C_n})$$

Proof. For a proof see [30, Lemma 4.1]

Standard Quadratic Programs

A special case of $SQC$ is the standard quadratic optimization problem (SQP). If we take $A = E_n = uu^T$, where $u$ is a vector consisting of all ones, and $b = 1$, and consider the fact $(u^T x)^T e^T x = 1$ if and only if $u^T x = 1$, then the standard quadratic optimization problem over the simplex will be,

$$\min_x \ x^T Qx$$

subject to

$$\begin{align*}
\text{SQP} & \quad u^T x = 1 \\
& \quad x \geq 0
\end{align*}$$

its copositive programming formulation will be,

$$\min_X \ \langle Q, X \rangle$$

subject to

$$\begin{align*}
\text{SQP}_{C_n^*} & \quad \langle E_n, X \rangle = 1 \\
& \quad X \in C_n^*
\end{align*}$$
where $E_n = u^Tu$ is $n$–dimensional matrix. The corresponding dual formulation,

$$\max_y y$$

s.t.

$$SQP_{C_n} \quad Q - yE_n \in C_n$$

From Lemma 4.1, we can deduce, 

**Corollary 4.2.** The optimization problems $SQP, SQP_{C_n}$ and $SQP_{C_n}$ are equivalent i.e.

$$\text{val}(SQP) = \text{val}(SQP_{C_n}) = \text{val}(SQP_{C_n})$$

**Remark** 1. It is interesting to note that the copositive and completely positive formulation of the standard quadratic optimization problem is the first celebrated result for copositive programming. Bomze et al [24] has given a polynomial time approximation scheme for the solution of the standard quadratic optimization problem for details see [24, 25].

2. Bomze and Schachinger [27], have also established equivalence for multi standard quadratic programming. In multi standard quadratic programming, quadratic function is maximized over the Cartesian product of several standard simplices. The simplicies can differ in dimension, for details see [27, 30].

### 4.2.2 Quadratically Constrained Quadratic Program

Recall from subsection 3.2.2

$$\min_X \langle Q, X \rangle + 2(c^0)^T x$$

s.t.

$$QP \quad \langle A_i, X \rangle + 2(c^i)^T x + b_i \leq 0, \quad \forall \ i \in \mathbb{I}$$

$$X = xx^T$$

$$x \geq 0$$
where $\mathbb{I} = \{1, \ldots, m\}, c^i \in \mathbb{R}^n, b_i \in \mathbb{R}$, consider the set $\mathcal{P}$,

$$\mathcal{P} = \left\{ P_i = \begin{pmatrix} b_i & (c^i)^T \\ c^i & A_i \end{pmatrix} \right\} \forall \ i \in \mathbb{I}$$

The set of feasible points of $QP$ is,

$$Feas(QP) = \left\{ x \in \mathbb{R}^n : \left\langle P_i, \begin{pmatrix} 1 \\ x \\ xx^T \end{pmatrix} \right\rangle \leq 0 \forall P_i \in \mathcal{P} \right\}$$

Here, instead of taking the SDP relaxation we will consider the copositive relaxation, described in the program given below,

$$\min_X \langle Q, X \rangle + 2(c^0)^T x$$

$$\text{s.t.}$$

$$QP_{CP} \quad \left\langle P_i, \begin{pmatrix} 1 \\ x \\ xx^T \end{pmatrix} \right\rangle \leq 0 \forall \ i \in \mathbb{I}$$

$$\left( \begin{array}{c} 1 \\ x \\ X \end{array} \right) \in C^*_n+1$$

Then the set of feasible points for the above program can be described by,

$$Feas(QP_{CP}) = \left\{ x \in \mathbb{R}^n : \exists X \in S_n \text{ such that } \left( \begin{array}{c} 1 \\ x \\ X \end{array} \right) \in C^*_n+1 \right. \left( \begin{array}{c} 1 \\ x \\ X \end{array} \right) \leq 0 \forall \ i \in \mathbb{I}$$

Let us denote by $Q'_+$, all copositive quadratic functions i.e.

$$Q'_+ = \{ x^T Ax + c^T x + \gamma : A \in C_n, c \in \mathbb{R}^n, \gamma \in \mathbb{R} \}$$

Let $\mathcal{K}$ denotes the convex cone generated by the quadratic functions $x^T A_i x + 2(c^i)^T x + b_i$ i.e.

$$\mathcal{K} = cone \{ \mathcal{P} \}$$
4.2. APPLICATIONS

Then we have following relaxation of $QP$

$$\begin{align*}
\min_x & \quad x^T Q x + 2 c^T x \\
\text{s.t.} & \quad \langle P, \begin{pmatrix} 1 & x^T \\ x & xx^T \end{pmatrix} \rangle \leq 0 \\
& \quad P \in K \cap Q'_+ \end{align*}$$

The next theorem describes the relation between these two relaxations,

**Theorem 4.3.**

$$\text{Feas}(QP_S) \subset \text{Feas}(QP_{CP})$$

**Proof.** One can define the dual cone of $K$ as,

$$K^* = \{ V \in S_n : \langle V, U \rangle \geq 0, \; \forall U \in K \}$$

Let $Q'_+^*$ is the dual of $Q'_+$, then by the definition of dual,

$$\begin{pmatrix} a & \pi^T \\ \pi & \Pi \end{pmatrix} \in Q'_+^* \iff \left\langle \begin{pmatrix} a & \pi^T \\ \pi & \Pi \end{pmatrix}, \begin{pmatrix} \gamma & c^T \\ c & A \end{pmatrix} \right\rangle \geq 0 \; \forall \begin{pmatrix} \gamma & c^T \\ c & A \end{pmatrix} \in Q'_+$$

From above we get,

$$\left\langle \begin{pmatrix} a & \pi^T \\ \pi & \Pi \end{pmatrix}, \begin{pmatrix} \gamma & c^T \\ c & A \end{pmatrix} \right\rangle \geq 0 \Rightarrow a\gamma + 2c^T \pi + \langle \Pi, A \rangle \geq 0$$

$$\Rightarrow a = 0, \pi = 0 \text{ and } \Pi \in C^*_n$$

hence $Q'_+^* = \begin{pmatrix} 0 & 0^T \\ 0 & C^*_n \end{pmatrix}$.
We can see that for the set of feasible points for $QP_S$ and $QP_{CP}$ we have,

$$Feas(QP_{CP}) = \left\{ x \in \mathbb{R}^n : \exists X \in S_n \text{ such that } \begin{pmatrix} 1 & x^T \\ x & X \end{pmatrix} \in (-K^* \cap C_{n+1}^*) \right\}$$

$$Feas(QP_S) = \left\{ x \in \mathbb{R}^n : \begin{pmatrix} 1 & x^T \\ x & xx^T \end{pmatrix} \in - (K \cap Q'_+) \right\}$$

$$= \left\{ x \in \mathbb{R}^n : \begin{pmatrix} 1 & x^T \\ x & xx^T \end{pmatrix} \in - \left( K^* + \begin{pmatrix} 0 & 0 \\ 0 & C_n^* \end{pmatrix} \right) \right\}$$

$Feas(QP_S) \subset Feas(QP_{CP})$: Let $x \in Feas(QP_S)$ then there exists some $H \in C_n^*$ such that,

$$\begin{pmatrix} 1 & x^T \\ x & xx^T + H \end{pmatrix} \in -K^*$$

Since $xx^T + H - xx^T = H \in C_n^*$ and $x \geq 0$ so Lemma 2.21 implies $\begin{pmatrix} 1 & x^T \\ x & xx^T + H \end{pmatrix} \in C_{n+1}^*$. Hence,

$$\begin{pmatrix} 1 & x^T \\ x & xx^T + H \end{pmatrix} \in -K^* \cap C_{n+1}^*$$

establishing, $x \in Feas(QP_{CP})$.

This completes the proof.

4.2.3 Stable Set Problem

Let $G = (V, E)$ be a graph with $n$ vertices. Then a stable set of the graph is defined as a set of vertices which are not adjacent to each other. The problem of finding the maximum stable set (sets of maximum cardinality such that no two vertices in the sets are adjacent) is NP-hard. The complement of the stable set problem is the clique problem. In clique problems it is required to find the set of vertices in the graph such that the subgraph is complete. Clique problem is also NP-hard and its decision version is NP-complete.

The stability number $\alpha(G)$ of a graph is defined as the cardinality of a maximum stable set in graph $G$. Clique number $\omega(\overline{G})$ is the cardinality of a maximum clique in the complement graph $\overline{G}$ (in the complement graph $\overline{G}$ of $G$, two vertices are adjacent if and only if they are not adjacent in $G$).

Another problem which can be associated with the above two problems is the $k$-coloring
4.2. APPLICATIONS

problem which is defined as: given a graph $G$ and an integer $k \geq 1$ then $k$-coloring is the assignment of $k$-colors to the vertices of $G$ such that adjacent vertices have distinct colors. The chromatic number or coloring number $\chi(G)$ of a graph is the smallest integer $k$ for which $G$ has a $k$-coloring. It is interesting to note that $k$-coloring gives the partition of vertices into $k$ stable sets. The three problems have the following relation,

$$\alpha(\bar{G}) = \omega(G) \leq \chi(G)$$

where $\bar{G} = (V, \bar{E})$ denote the complementary graph of $G$. For the mathematical programming formulation of the stable set problem we associate with any stable set $S \subset V$ a characteristic vector $x_S \in \{0, 1\}^n$ defined as,

$$(x_S)_v = \begin{cases} 
1 & v \in S \\
0 & v \notin S 
\end{cases}$$

On the basis of the characteristic vector we can define a matrix $X$ by,

$$X = \frac{1}{x_S^Tx_S} x_Sx_S^T$$

which has the property that $|S| = \langle X, J \rangle$, moreover $X \succeq 0$, $X_{ij} = 0, \; \forall (i,j) \in E$ and $\langle I, X \rangle = \text{trace}(X) = 1$, resulting in the following semidefinite relaxation of the maximum stable set problem (see Lovasz [102]),

$$\max \langle J, X \rangle$$

s.t.

$$MC_{SDP} \quad X_{ij} = 0 \; \forall \; i \neq j \; \text{and} \; (i,j) \in E$$

$$\langle I, X \rangle = 1$$

$$X \in S_n^+$$

where $J$ is a matrix consisting of all ones. The optimum value of the above program gives the so called Lovasz theta number i.e $\vartheta(G) = \text{val}(MC_{SDP})$. Moreover $\vartheta(G)$ gives a bound on the stability and the chromatic number of graphs,

**Theorem 4.4.** For any given graph $G$, one has

$$\alpha(G) \leq \vartheta(G) \leq \chi(\bar{G})$$
4.2. APPLICATIONS

Remark A graph is called perfect if $\omega(G') = \chi(G')$ for every induced subgraph $G'$ of $G$. The existence of polynomial time algorithm for the stability and chromatic number of perfect graphs became possible only after the definition of the theta number. The inequalities given in the above theorem become strict equalities in case of perfect graphs (see [94, 102]).

Having $MC_{SDP}$ it seems natural to consider a copositive relaxation of this problem. de Klerk and Pasechnik [45] considered the following copositive relaxation of $MC_{SDP}$,

$$\max \langle J, X \rangle$$
$$s.t.$$  
$$MC_{CP} X_{ij} = 0 \quad \forall \ i \neq j \text{ and } (i, j) \in E$$  
$$\langle I, X \rangle = 1$$  
$$X \in C^*_n$$

**Theorem 4.5.** Let $G(V, E)$ be given with $|E| = n$. The stability number of $G$ is given by $val(MC_{CP})$ i.e $\alpha(G) = val(MC_{CP})$.

**Proof.** For proof see [45, Theorem 2.2]. \hfill \Box

Since $X \in C^*_n$ implies $X$ is nonnegative. So $\langle I, AX \rangle = 0$, where $A = \{a_{ij}\}$ is the adjacency matrix of the graph $G$, if and only if $X_{ij} = 0$, $\forall \ i \neq j$ and $(i, j) \in E$. Hence we can write the above program as follows,

$$\max \langle J, X \rangle$$
$$s.t.$$  
$$MC_{CP} \quad \langle I, AX \rangle = 0$$  
$$\langle I, X \rangle = 1$$  
$$X \in C^*_n$$

Now we can write the above program equivalently in the following form (for details see
\[ \text{max} \quad \langle J, X \rangle \]
\[ \text{s.t.} \]
\[ MC_{CP} \quad \langle (A + I)^T, X \rangle = 1 \]
\[ X \in C_n^* \]

Remark The copositive relaxation of the stability problem is first discussed by de Klerk and Pasechnick [45] in 2002. The chromatic and the stability number of a graph are perhaps the most studied problems with respect to copositive programming [17, 45, 51].

### 4.2.4 Graph Partitioning

Let \( G = (V, E) \) be an undirected graph with a set of vertices \( V \), having cardinality \( n \) and the set of edges \( E \) with weight \( a_{ij} \) for each edge. Then the 3-partitioning of vertices is the partition of the vertices into 3 disjoint sets \( S_1, S_2 \) and \( S_3 \) of specified cardinalities such that the total weight of all edges joining \( S_1 \) and \( S_2 \) is minimized. We use the symmetric nonnegative matrix \( A \) called adjacency matrix of graph, with \( A = \{a_{ij}\} \) and \( a_{ij} > 0 \) if there exists an edge \( e_{ij} \in E(G) \) (with weight \( a_{ij} \)), where \( e_{ij} \) denote the edge between vertices \( i \) and \( j \). If we denote the total weight of edges between sets \( S_1 \) and \( S_2 \) as \( \text{cut}(S_1, S_2) \), then we have the optimization problem,

\[ \min \quad \text{cut}(S_1, S_2) = \sum_{i \in S_1, j \in S_2} a_{ij} \]

\[ GP_0 \quad (S_1, S_2, S_3) \text{ partitions of } V(G) \]

\[ |S_i| = m_i, \quad i = 1, 2, 3 \]

Let \( X \in \{0, 1\}^{(n \times 3)} \) be a matrix whose \( j^{th} \) columns is a vector \( x_j \) of the form,

\[ (x_j)_i = x_{ij} = \begin{cases} 
1 & \text{if } i \in S_j \\
0 & \text{if } i \notin S_j 
\end{cases} \quad \forall j = 1, 2, 3 \]
4.2. APPLICATIONS

then

\[ \sum_{i \in S_1, j \in S_2} a_{ij} = x_1^T A x_2 = \text{trace}(A x_1^T x_2^T) = \frac{1}{2} \text{trace}(A X B X^T) = \frac{1}{2} \langle X, AXB \rangle \]

where \( B = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \). Then \( GP_0 \) can be written as,

\[
\begin{align*}
\min & \quad \frac{1}{2} \langle X, AXB \rangle \\
\text{s.t.} & \quad X^T X = \text{Diag}(m) \\
& \quad X u_3 = u_n \\
& \quad X \in \{0, 1\}^{(n \times 3)}
\end{align*}
\]

where \( m = (m_1, m_2, m_3)^T \) is a vector such that \( m_i = |S_i|, i = 1, 2, 3 \).

A usual way to make quadratic program of the above form convex is to gather all nonlinearities in a single constraint. Hence obtaining a program which has linear constraints with exception of one nonlinear constraint of the form \( Y = xx^T \). In case of \( GP_0 \), we will first make the linear constraint quadratic and then we will linearize the quadratic terms, which result in gathering all linearities in the constraint \( Y = xx^T \). Following is an equivalent formulation of the above program,

\[
\begin{align*}
\min & \quad \frac{1}{2} \langle B^T \otimes A, Y \rangle \\
\text{s.t.} & \quad \langle B_{ij} \otimes I, Y \rangle = m_i \delta_{ij}, \quad \forall \quad 1 \leq i \leq j \leq 3 \\
& \quad \langle J_3 \otimes E_{ii}, Y \rangle = 1, \quad \forall \quad 1 \leq i \leq n \\
& \quad \langle V_i \otimes W_j^T, Y \rangle = m_i, \quad \forall \quad 1 \leq i \leq 3, \ 1 \leq j \leq n \\
& \quad \langle B_{ij} \otimes J_n, Y \rangle = m_i m_j, \quad \forall \quad 1 \leq i \leq j \leq 3 \\
& \quad Y = xx^T, \quad x \in \mathbb{R}^{3n}_+
\end{align*}
\]
where \( x = \text{vec}(X) \), \( E_{ij} = e_i e_j^T \), \( B_{ij} = \frac{1}{2}(E_{ij} + E_{ji}) \), \( V_i = e_i u_i^T \in \mathbb{R}^{3 \times 3} \), \( J_n \) is \( n \)-dimensional matrix consisting of only ones and \( W_i = e_i u_n^T \in \mathbb{R}^{n \times n} \), \( 1 \leq i \leq 3 \), \( 1 \leq j \leq n \).

Just like SDP relaxation, the rank one matrix constraint \( Y = xx^T \) is relaxed by the copositive constraint \( Y \in C_{3n}^* \), hence we will have,

\[
\min_Y \frac{1}{2} \langle B^T \otimes A, Y \rangle \\
\text{s.t.} \\
\langle B_{ij} \otimes I, Y \rangle = m_i \delta_{ij}, \quad \forall \ 1 \leq i \leq j \leq 3 \\
\langle J_3 \otimes E_{ii}, Y \rangle = 1, \quad \forall \ 1 \leq i \leq n \\
\langle V_i \otimes W_j^T, Y \rangle = m_i, \quad \forall \ 1 \leq i \leq 3, \ 1 \leq j \leq n \\
\langle B_{ij} \otimes J_n, Y \rangle = m_i m_j, \quad \forall \ 1 \leq i \leq j \leq 3 \\
Y \in C_{3n}^*
\]

Povh and Rendl [116] proved that this relaxation of the partition problem is exact,

**Theorem 4.6.** Let \( x \in \mathbb{R}^{3n}_+ \) and \( Y \in \text{Feas}(GP_{CP}) \) then

\[
\text{Conv}\{xx^T; Y = xx^T \in \text{Feas}(GP)\} = \text{Feas}(GP_{CP})
\]

**Proof.** see [116].

The immediate consequence of this theorem is the following corollary,

**Corollary 4.7.** \( \text{val}(GP) = \text{val}(GP_{CP}) \)

**Remark**

1. In \( GP \) if we replace \( Y = xx^T \) by \( Y \in S_n^+ \) then we obtain an SDP relaxation.

Povh and Rendl [116] have proved that their proposed relaxation is equivalent to another SDP relaxation proposed by Helmberg et al [76] (replace the constraint \( X \in \{0, 1\}^{(n \times 3)} \) with the constraint of the form \( X^T u_n = m \)). SDP relaxation proposed by Helmberg et al [76] is mainly used to obtain easy to compute lower bounds for graph partitioning problems.

2. SDP relaxation of the partition problem is not only discussed by Povh and Rendl. As a matter of fact there exist semidefinite relaxations of the more general \( k \)-partition problem in which we divide vertices into \( k \) disjoint sets instead of only three sets, for details see [61, 76, 144, 146].
4.2 APPLICATIONS

4.2.5 Quadratic Assignment Problem (QAP)

QAP is one of the most studied problems in combinatorial optimization. It was first introduced by Koopmans and Beckmann [89] as facility location problem. Nowadays the application of QAP is not restricted to facility location problem only. QAP has been applied to other areas like wiring problems in electronics, scheduling, parallel and distributed computing, statistical data analysis, design of control panel, chemistry, archeology and design of turbine runners etc [38].

In QAP you are given a set of \( n \) location and \( n \) facilities and the distance between these locations. The objective of this problem is to find assignments of locations to facilities such that the cost is minimized. Normally a permutation vector denoted by \( \pi \) is used to denote the assignment. Koopmans and Beckmans presented this problem as:

\[
\text{min} \sum_{i,j} a_{ij} b_{\pi(i)\pi(j)} + \sum_i c_{i,\pi(i)}
\]

s.t.

\( \pi \) is a permutation of \( \{1, ..., n\} \)

where \( A = \{a_{ij}\}, B = \{b_{ij}\}, C = \{c_{ij}\} \in \mathbb{R}^{n \times n} \). Each permutation can be represented by the permutation matrix \( X = \{x_{ij}\} \), with,

\[
x_{ij} = \begin{cases} 
1 & \text{if } \pi(i) = j \\
0 & \text{otherwise}
\end{cases}
\]

Let \( \Pi \) be the set of all permutation matrices \( X \). Then QAP can be written as,

\[
\text{min} \quad \langle X, AXB + C \rangle \\
\text{s.t.} \\
X \in \Pi,
\]

It can be seen easily that every permutation matrix is orthonormal since there is only one
4.2. APPLICATIONS

nonzero term in each row and column of the permutation matrix, so we can write,

$$\Pi = \{ X \in \mathbb{R}^{n \times n} : X^T X = I, X \geq 0 \}$$

then,

$$\min \langle X, AXB + C \rangle$$

s.t.
$$X^T X = I,$$
$$X \geq 0$$

If we add an extra redundant constraint $$XX^T = I$$ in the above program then we obtain,

$$\min_X \langle X, AXB \rangle$$

s.t.
$$QAP \quad X^T X = I,$$
$$XX^T = I,$$
$$X \geq 0$$

Let $$Y$$ and $$Z$$ be two symmetric matrices corresponding to the Lagrangian functions of the constraints $$X^T X = I$$ and $$XX^T = I$$ respectively (for complete details see [8, 94]). Then we obtain the following program,

$$\max \langle I, Y + Z \rangle$$

s.t.
$$\text{QAP}_{LD} \quad (I \otimes Y) + (I \otimes Z) \preceq (B \otimes A)$$
$$Y^T = Y, Z^T = Z$$

The optimum value of QAP and QAP_{LD} is equal,

**Theorem 4.8.** $$\text{val}(QAP) = \text{val}(QAP_{LD})$$ giving that the strong duality holds for the pair QAP and QAP_{LD}.

*Proof.* see [8].

\[\square\]
4.2. APPLICATIONS

Povh and Rendl [117], has extended the ideas by giving a copositive relaxation. They have added one more redundant constraint by considering the fact that sum of all elements in $X$ is $n$, hence they have included the constraint $\langle X, JXJ \rangle = (u^T X u)^2 = n^2$. By adding this constraint QAP will become,

$$
\begin{align*}
\min & \quad \langle X, AXB + C \rangle \\
\text{s.t.} & \quad X^T X = I \\
& \quad XX^T = I \\
& \quad \langle X, JXJ \rangle = n^2 \\
& \quad X \geq 0 \\
\end{align*}
$$

Just like $QAP_{LD}$ we introduce two matrices $Y, Z$ and $v \in \mathbb{R}$, the Lagrange variables corresponding to the constraints $X^T X = I, XX^T$ and $\langle X, JXJ \rangle = n^2$ respectively. Moreover consider the fact, $\langle C, X \rangle = \langle Diag(c), xx^T \rangle$ for $X \in \Pi$ and $\langle X, PXQ \rangle = \langle Q \otimes P, xx^T \rangle$ for any $X \geq 0$ where $x = vec(X), c = vec(C)$ implies $\langle X, AXB + C \rangle = \langle B \otimes A + Diag(c), Y \rangle$. We obtain copositive relaxation of $QAP_P$ given below,

$$
\begin{align*}
\max_{Z,Y,v} & \quad \langle I, Y + Z \rangle + n^2 v \\
\text{s.t.} & \quad QAP_{P(C)} \\
& \quad B \otimes A + Diag(c) - I \otimes Y - Z \otimes I - vJ_n \in C_{n^2} \\
& \quad Z, Y \in S_n \\
\end{align*}
$$
The dual of above program is given by,

\[ \min \langle B \otimes A + \text{Diag}(c), T \rangle \]

s.t.

\[ \sum_i T^{ii} = I \]

\[ \langle I, T^{ij} \rangle = \delta_{ij} \forall i, j \]

\[ \langle J_{n^2}, T \rangle = n^2 \]

\[ T \in C^*_{n^2} \]

where \( T = \begin{pmatrix} T^{11} & \cdots & T^{1n} \\ \vdots & \ddots & \vdots \\ T^{n1} & \cdots & T^{nn} \end{pmatrix} \) with \( T^{ij} \in \mathbb{R}^{n \times n} \).

It is interesting to note that strong duality hold for the primal dual pair \( QAP_P \) and \( QAP_{P(C)} \). Since \( QAP_{P(C)} \), is strictly feasible (for \( Z = Y = -\alpha I \) and \( v = 0 \) the matrix \( B \otimes A + \text{Diag}(c) + 2\alpha I_{n^2} \) is strictly copositive). Povh and Rendl have established the equivalence of \( QAP_P \) and \( QAP_{P_D} \), see Theorem 4.9.

**Theorem 4.9.** \( \text{val}(QAP_P) = \text{val}(QAP_{P_D}) \)

**Proof.** see [117].

### 4.2.6 Mixed Binary Quadratic Programming

Recently Burer [34], has given a completely positive relaxation of a mixed quadratic binary program. Burer has considered the program,

\[ \min_x x^T Q x + 2c^T x \]

s.t.

\[ a_i^T x = b_i, \ \forall i = 1, \ldots, m \]

\[ QP_B \]

\[ x \geq 0 \]

\[ x_j \in \{0,1\} \ \forall \ j \in \mathbb{I} \]
4.2. APPLICATIONS

where $x \in \mathbb{R}^n$ and $\mathbb{I} \subseteq \{1, ..., n\}$. Consider $(a_i^T x)^2 = b_i^2 \iff a_i^T x x^T a_i = b_i^2$, we will add these constraints in $QP_B$. It is well known \[55, 94\], that the condition $x_j \in \{0, 1\}$ can be equivalently stated as $x_j^2 - x_j = 0$. If $X = x x^T$ then $x_j^2 - x_j = 0$ implies $X_{jj} = x_j$. Just like in SDP we will relax the rank one matrix constraints $X = x x^T$, $x \geq 0$ by the constraint

\[
\begin{pmatrix}
1 \\ x^T \\
X
\end{pmatrix} \in C_{n+1}^* \text{ (since } X = x x^T \text{ if and only if } \begin{pmatrix}
1 \\ x^T \\
X
\end{pmatrix} = \begin{pmatrix}
1 \\ 1
\end{pmatrix} \begin{pmatrix}
1 \\ x^T \\
X
\end{pmatrix}^T).
\]

Then we will have the relaxation,

\[
\min_{X, x} \langle Q, X \rangle + 2c^T x
\]
\[
\text{s.t.}
\]
\[
CPP_B \quad a_i^T x = b_i, \quad \forall \ i = 1, ..., m
\]
\[
a_i^T X a_i = b_i^2 \quad \forall \ i = 1, ..., m
\]
\[
X_{jj} = x_j \quad \forall \ j \in \mathbb{I}
\]
\[
\begin{pmatrix}
1 \\ x^T \\
X
\end{pmatrix} \in C_{n+1}^*
\]

In order to derive a relationship among the set of feasible points of $QP_B$ and $CPP_B$ we make the following assumption,

**Assumption 4.10.** Let $L = \{ x \geq 0 : a_i^T x = b_i \ \forall i = 1, ..., m \}$ be the linear part of $\text{Feas}(QP_B)$ then

\[x \in L \quad \Rightarrow \quad 0 \leq x_j \leq 1 \ \forall \ j \in \mathbb{I}\]

**Remark**

1. It is clear that if Assumption \[4.10\] does not hold then it can be implied by introducing at most $|\mathbb{I}|$ slack variables $z_j$ such that $x_j + z_j = 1$.

2. Bomze and Jerry \[18\] has noted that the above assumption can be replaced by a weaker assumption namely

**Assumption 4.11.**

\[x \in L \quad \Rightarrow \quad x_j \text{ is bounded for all } j \in \mathbb{I}\]

This weaker assumption can be enforced by introducing just one slack variable instead of $|\mathbb{I}|$ slack variables.
The recession cone of $L$ is given by,

$$L_{\infty} = \{ d \geq 0 : a_i^T d = 0 \ \forall i = 1, ..., m \}$$

then it is clear from the Assumption 4.10 that if $L \neq \emptyset$ then,

$$d \in L_{\infty} \Rightarrow d_j = 0 \ \forall j \in \mathbb{I} \quad (4.1)$$

In order to show that the relaxation of $QP_B$ is indeed exact, we will consider the new feasible set namely $Feas^+(CPP_B)$ and $Feas^+(QP_B)$ which are linear parametrization of $Feas(CPP_B)$ and $Feas(QP_B)$ respectively, and are defined by:

$$Feas^+(CPP_B) = \left\{ \begin{pmatrix} 1 & x^T \\ x & X \end{pmatrix} : (x, X) \in Feas(CPP_B) \right\}$$

$$Feas^+(QP_B) = \text{Conv} \left\{ \begin{pmatrix} 1 \\ x \end{pmatrix} \begin{pmatrix} 1 \\ x \end{pmatrix}^T : x \in Feas(QP_B) \right\}$$

$$L_{\infty}^+ = \text{Cone} \left\{ \begin{pmatrix} 0 \\ d \end{pmatrix} \begin{pmatrix} 0 \\ d \end{pmatrix}^T : d \in L_{\infty} \right\} \quad (4.2)$$

Theorem 4.12 gives the relationship between the above sets of feasible points,

**Theorem 4.12.** $Feas^+(CPP_B) = Feas^+(QP_B) + L_{\infty}^+$

**Proof.** $Feas^+(CPP_B) \subseteq Feas^+(P) + L_{\infty}^+$: For this we will prove that for any $(x, X) \in Feas(CPP_B)$ we have a decomposition of the form,

$$\begin{pmatrix} 1 & x^T \\ x & X \end{pmatrix} = \sum_{k \in \Phi_+} \lambda_k \begin{pmatrix} 1 \\ v_k \end{pmatrix} \begin{pmatrix} 1 \\ v_k \end{pmatrix}^T + \sum_{k \in \Phi_0} \begin{pmatrix} 0 \\ y_k \end{pmatrix} \begin{pmatrix} 0 \\ y_k \end{pmatrix}^T \quad (4.3)$$

where $\sum_{k \in \Phi} \lambda_k = 1$, $v_k \in L \ \forall k \in \Phi_+$ and $y^k \in L_{\infty} \ \forall k \in \Phi_0$.

Let $(x, X) \in Feas(CPP_B)$ and consider the completely positive decomposition,

$$\begin{pmatrix} 1 & x^T \\ x & X \end{pmatrix} = \sum_{i \in \Phi} \begin{pmatrix} \xi_k \\ y^k \end{pmatrix} \begin{pmatrix} \xi_k \\ y^k \end{pmatrix}^T \quad (4.4)$$
4.2. APPLICATIONS

with \( y^k \in \mathbb{R}_+^n, \xi_k \in \mathbb{R}_+ \). We define, \( \Phi_+ = \{ k \in \Phi : \xi_k > 0 \} \) and \( \Phi_0 = \{ k \in \Phi : \xi_k = 0 \} \).

From equation (4.4), we will have,

\[
\left( \begin{array}{c}
1 \\
x \\
X
\end{array} \right) = \sum_{k \in \Phi} \left( \begin{array}{c}
\xi_k \\
y^k
\end{array} \right) \left( \begin{array}{c}
\xi_k \\
y^k
\end{array} \right)^T = \sum_{k \in \Phi} \left( \begin{array}{cc}
\xi_k^2 & \xi_k y^k y^k^T \\
\xi_k y^k & y^k y^k^T
\end{array} \right)
\]

(4.5)

hence,

\[
\sum_{k \in \Phi} \xi_k^2 = 1.
\]

(4.6)

We identify \( \lambda_k, k \in \Phi_+ \) by \( \xi_k^2 \), hence giving \( \sum_{k \in \Phi_+} \lambda_k = 1 \).

From \( b_i = a_i^T x \) and \( b_i^2 = a_i^T Xa_i \), and (4.5), one can see,

\[
b_i = a_i^T x = \sum_{k \in \Phi} \xi_k y^k \sum_{k \in \Phi} \xi_k (a_i^T y^k) \quad \forall \ i = 1, \ldots, m
\]

(4.7)

and

\[
b_i^2 = a_i^T Xa_i = \sum_{k \in \Phi} y^k y^k^T a_i \sum_{k \in \Phi} \xi_k^2 (a_i^T y^k)^2 \quad \forall \ i = 1, \ldots, m
\]

(4.8)

So we can write, from (4.6), (4.7), (4.8)

\[
\left( \sum_{k \in \Phi} \xi_k (a_i^T y^k) \right)^2 = \left( \sum_{k \in \Phi} \xi_k^2 \right) \left( \sum_{k \in \Phi} (a_i^T y^k)^2 \right)
\]

In Cauchy-Schwarz inequality, equality holds if and only if the two vectors \((\xi_k : k \in \Phi)\) and \((a_i^T y^k : k \in \Phi)\) are linearly dependent, so in our case for all \( i = 1, \ldots, m \), there exists \( \delta_i \) such that,

\[
\delta_i \xi_k = a_i^T y^k \quad \forall \ k \in \Phi, \ i = 1, \ldots, m
\]

(4.9)

If \( k \in \Phi_0 \) then \( \xi_k = 0 \) than (4.9), will become

\[
a_i^T y^k = 0 \quad \forall \ k \in \Phi_0, \ i = 1, \ldots, m
\]

hence \( y^k \in L_\infty, \forall \ k \in \Phi_0 \). Consider (4.7),

\[
b_i = \sum_{k \in \Phi} \xi_k (a_i^T y^k) = \sum_{k \in \Phi} \xi_k (\delta_i \xi_k) = \delta_i \sum_{k \in \Phi} (\xi_k)^2 = \delta_i
\]

53
So for all \( k \in \Phi_+ \), we will have

\[
\delta_i \xi_k = a_i^T y^k \Rightarrow b_i = a_i^T \frac{y^k}{\xi_k} \forall k \in \Phi_+ \Rightarrow \frac{y^k}{\xi_k} \in L \forall k \in \Phi_+
\]

So we can conclude that \( v^k := \frac{y^k}{\xi_k} \) satisfies the linear constraints.

For the binary constraint, let \( j \in \mathbb{I} \). From constraint \( X_{jj} = x_j \), we will get,

\[
\sum_{k \in \Phi_+} \lambda_k (v_j^k) = \sum_{k \in \Phi_+} \lambda_k (v_j^k)^2 + \sum_{k \in \Phi_0} (\frac{y_j^k}{\xi_k})^2
\]

\[
\iff \sum_{k \in \Phi_+} \lambda_k (v_j^k - (v_j^k)^2) = 0
\]

In above expression \( y_j^k = 0 \), since (4.3) implies \( y^k \in L_{\infty} \) and from (4.1) we get \( y_j^k = 0 \) for all \( j \in \mathbb{I} \). Moreover, since \( \lambda_k > 0 \) and \( 0 \leq v_j^k \leq 1, \forall j \in \mathbb{I} \), so the sum of nonnegative numbers is zero if and only if each one of them is zero that is \( v_j^k - (v_j^k)^2 = 0 \), giving that \( v_j^k \in \{0,1\} \ \forall j \in \mathbb{I} \). Hence the binary constraints are satisfied for all \( j \in \mathbb{I} \). So decomposition of the from (4.3) exists and the inclusion follows.

\textit{Feas}^+(QP_B) + L_{\infty}^+ \subseteq \textit{Feas}^+(CPP_B): \textit{Consider} \textit{Feas}^+(QP_B) \subseteq \textit{Feas}^+(CPP_B) \textit{which follows directly from the construction. The recession cone (see Definition 2.7) of} \textit{Feas}^+(CPP_B) \textit{given by,}

\[
\text{rec}(\text{Feas}^+(CPP_B)) = \left\{ \begin{align*}
\begin{pmatrix} 0 & d^T \\ d & D \end{pmatrix} & \in C_{1+n}^* : \begin{align*}
a_i^T d &= 0 \quad i = 1, \ldots, m \\
a_i^T D a_i &= 0 \quad i = 1, \ldots, m \\
d_j &= D_{jj} \quad j \in \mathbb{I}
\end{align*}
\end{pmatrix}
\right\}
\]

Since \( \begin{pmatrix} 0 & d^T \\ d & D \end{pmatrix} \in C_{1+n}^* \), it should have a decomposition of the form (4.3) giving \( d = 0 \), and the recession cone then simplifies to,

\[
\text{rec}(\text{Feas}^+(CPP_B)) = \left\{ \begin{pmatrix} 0 & 0^T \\ 0 & D \end{pmatrix} \in C_{1+n}^* : \begin{align*}
a_i^T D a_i &= 0 \quad i = 1, \ldots, m \\
D_{jj} &= 0 \quad j \in \mathbb{I}
\end{align*}
\right\}
\]

54
From (4.1) and (4.2) we can write,

\[ \mathcal{L}_\infty^+ = \text{Cone} \left\{ \begin{pmatrix} 0 \\ d \end{pmatrix} : d \in \mathcal{L}_\infty \right\} = \text{cone} \left\{ \begin{pmatrix} 0 & 0^T \\ 0 & dd^T \end{pmatrix} : d \in \mathcal{L}_\infty \right\} \]

which shows that \( \mathcal{L}_\infty^+ \) is contained in the recession cone of \( \text{Feas}^+(\text{CPP}_B) \) so the inclusion follows.

Hence the proof is complete.

The following is an immediate consequence of Theorem 4.12.

**Theorem 4.13.** Under Assumption 4.10, we have,

1. \( \text{val} (\text{CPP}_B) = \text{val} (\text{QP}_B) \)

2. If \( (x^*, X^*) \) is optimal for \( \text{CPP}_B \), then \( x^* \) is in the convex hull of optimal solutions for \( \text{QP}_B \).

**Proof.** 1. Let us define the linear function,

\[ l(Y) = \left\langle \begin{pmatrix} 0 & c^T \\ c & Q^T \end{pmatrix} Y \right\rangle \quad \text{(4.10)} \]

It is not difficult to verify that the optimum values of \( \text{QP}_B \) and \( \text{CPP}_B \) can be written in terms of \( l(Y) \) in following manner,

\[ \text{val}(\text{QP}_B) = \min_{Y \in \text{Feas}^+(\text{QP}_B)} l(Y) \quad \text{and} \quad \text{val}(\text{CPP}_B) = \min_{Y \in \text{Feas}^+(\text{CPP}_B)} l(Y) \]

Since \( \text{Feas}^+(\text{QP}_B) \subseteq \text{Feas}^+(\text{CPP}_B) \), hence we will have \( \text{val}(\text{QP}_B) \geq \text{val}(\text{CPP}_B) \).

In order to obtain equality we will consider following two cases,

\( \text{val}(\text{QP}_B) = -\infty \): If \( \text{val}(\text{QP}_B) = -\infty \) then from \( \text{val}(\text{QP}_B) \geq \text{val}(\text{CPP}_B) \), we obtain that \( \text{val}(\text{CPP}_B) = -\infty \), hence \( \text{val}(\text{QP}_B) = \text{val}(\text{CPP}_B) \). So in case of infinite values the optimum values are same.

\( \text{val}(\text{QP}_B) \neq -\infty \): From Theorem 4.12 we have,

\[ \text{Feas}^+(\text{CPP}_B) = \text{Feas}^+(\text{QP}_B) + L_\infty^+ \]
4.2. APPLICATIONS

The above equation suggests that if \( l \) is nonnegative over \( L_\infty^+ \) then optimum values of \( QP_B \) and \( CPP_B \) coincides with each other. In order to show that \( l \) is nonnegative over \( L_\infty^+ \) take \( d \in L_\infty \) then,

\[
l \left( \begin{pmatrix} 0 \\ d \end{pmatrix} \right) = \left( \begin{pmatrix} 0 & c^T \\ c & Q \end{pmatrix}, \begin{pmatrix} 0 \\ d \end{pmatrix} \right)^T = d^T Q d
\]

From above equation it is clear that \( l \) is negative only if \( d^T Q d \) is negative which implies that we can have optimum value from negative orthant hence from \( \text{val}(QP_B) = \min_{Y \in \text{Feas}(QP_B)} l(Y) \) we get \( \text{val}(QP_B) = -\infty \) which is a contradiction. Hence \( d^T Q d \) is nonnegative. Since \( d \in L_\infty \) is arbitrary so \( l \) is nonnegative on \( L_\infty \).

2. Since \( (x^*, X^*) \) is optimal for \( CPP_B \) hence it will have decomposition of the form \( [1.3] \) that is,

\[
\begin{pmatrix} 1 & (x^*)^T \\ x^* & X^* \end{pmatrix} = \sum_{k \in \Phi_+} \lambda_k \begin{pmatrix} 1 & (v^k)^T \\ v^k & v^k v^k^T \end{pmatrix} + \sum_{k \in \Phi_0} \begin{pmatrix} 0 \\ y^k \end{pmatrix} \begin{pmatrix} 0 \\ y^k \end{pmatrix}^T
\]

giving that, \( x^* = \sum_{k \in \Phi_+} \lambda_k v^k \). By Theorem \( [4.12] \) \( v^k \in \text{Feas}(QP_B) \) hence \( x^* \in \text{Cov}\{\text{Feas}(QP)\} \). Moreover \( v^k \) is also optimal for \( QP_B \) since,

\[
\text{val}(CPP_B) = l \left( \begin{pmatrix} 1 & (x^*)^T \\ x^* & X^* \end{pmatrix} \right) = \sum_{k \in \Phi_+} \lambda_k \langle Q, v^k(v^k)^T \rangle + 2c^T \sum_{k \in \Phi_+} \lambda_k v^k + \sum_{k \in \Phi_+} \lambda_k \langle Q, y^k(y^k)^T \rangle
\]

\[
\geq \text{val}(QP_B) = \text{val}(CPP_B)
\]

the last equality holds from part 1. Hence \( v^k \) is optimal and this completes the proof. \( \square \)

**Remark** Bomze and Jarre [18], have showed that under the weak assumption we will have \( \overline{\text{Feas}^+(QP_B)} = \text{Feas}^+(QP_B) + L_\infty^+ = \text{Feas}^+(CPP_B) \), where \( \overline{\text{Feas}^+(QP_B)} \) denotes the closure of \( \text{Feas}^+(QP_B) \).

A disadvantage of \( CPP_B \) is that it does not contain interior points. Since,

\[
\begin{pmatrix} b_i - a_i \\ x^T \\ a_i^T \end{pmatrix} \begin{pmatrix} 1 & x^T \\ x & X \end{pmatrix} \begin{pmatrix} b_i \\ a_i^T \end{pmatrix} = 0, \quad \forall \begin{pmatrix} 1 & x^T \\ x & X \end{pmatrix} \in \text{Feas}(CPP_B)
\]

56
Thus the matrix $\begin{pmatrix} 1 & x^T \\ x & X \end{pmatrix}$ is not in the interior of semidefinite cone so by definition of completely positive matrices the matrix $\begin{pmatrix} 1 & x^T \\ x & X \end{pmatrix}$ can not lie in the interior of the completely positive cone. Recall that the interior of feasible set of points for a certain cone program is crucial for obtaining strong duality results. Therefore nonexistence of interior points is a disadvantage. In order to solve this problem Burer, suggests to eliminate $x$ from the completely positive representation. Elimination of $x$ is possible only if there exists some $y \in \mathbb{R}^m$ such that,

$$\sum_{i=1}^{m} y_i a_i \geq 0 \quad \text{and} \quad \sum_{i=1}^{m} y_i b_i = 1$$

If we consider above constraints then we can write $CPP_B$ after elimination of $x$ as,

$$\min_X \langle Q, X \rangle + 2c^TX\alpha$$

s.t.

$$a_i^T X \alpha = b_i, \quad i = 1, \ldots, m$$

$$a_i^T X a_i = b_i^2, \quad i = 1, \ldots, m$$

$$CPP_{reduced} \quad (X\alpha)_k = X_{kk} \quad \forall \ k \in B$$

$$\alpha^T X \alpha = 1$$

$$X \in C_n^*$$

where,

$$\alpha = \sum_{i=1}^{m} y_i a_i \geq 0$$

$CPP_{reduced}$ is equivalent to $CPP_B$, as stated in next theorem.

**Theorem 4.14.** If there exists $y \in \mathbb{R}^m$ such that $\sum_{i=1}^{m} y_i a_i \geq 0$ and $\sum_{i=1}^{m} y_i b_i = 1$. We define $\alpha = \sum_{i=1}^{m} y_i a_i \geq 0$ then $CPP_{reduced}$ is equivalent to $QP_B$ i.e

1. $\text{opt}(CPP_{reduced}) = \text{opt}(CPP_B)$

2. If $X^*$ is optimal for $CPP_{reduced}$, then $X^* \alpha$ is in the convex hull of optimal solution for $CPP_B$
4.3 Algorithm

Copositive programming being a new area of research does not enjoy the luxury of having some standard solution method. Although copositive programming has been applied to a number of practical problems, still there is no algorithm for solving copositive programming. Some efforts have been made for developing a solution procedure. But these methods are not really efficient and have a number of deficiencies. Beside the fact that the available methods are not good enough, they are still applied to practical problems.

To the best of our knowledge there exists three methods/procedures for solving copositive programs. The oldest and most widely used method is based on an approximation of the copositive cone by different cone hierarchies. Another method which is based on inner and outer approximations of the copositive cone is also applied to certain practical problems. The feasible descent method of Bomze et al [26] provides a choice for solving copositive programs over the cone of completely positive matrices. In the next subsections we will briefly discuss each of the above three procedures/methods for solving copositive programs.

4.3.1 Approximation Hierarchy Based Methods

These methods are based on the quadratic form associated with copositive matrices. With every matrix $A$ we can associate a quadratic form. If this quadratic form is nonnegative for nonnegative argument then the matrix is copositive. This quadratic form can be written as finite sum of quadratic terms, with elements of $A$ being coefficient of the polynomial, hence for a certain matrix $A \in S_n$, we can define the polynomial,

$$P_A(x) = \sum_{i=1}^{n} \sum_{j=1}^{n} A_{ij} x_i x_j$$

It is clear that $A \in C_n$ if and only if $P_A(x) \geq 0$ for all $x \in \mathbb{R}^n$. So the problem of checking copositivity of a matrix reduces to checking conditions when $P_A(x)$ is nonnegative. It is not difficult to verify that $P_A(x)$ will be nonnegative if,

1. $P_A(x)$ can be written as sum of square
2. All the coefficients of $P_A(x)$ are nonnegative

Whenever the first condition holds the second condition become true automatically. It is shown by Parrilo [112] that $A \in S_n^+ + N_n$ if and only if $P_A(x)$ allows a sum of square
4.3. ALGORITHM

distribution. Moreover if $f(x_1, \ldots, x_n)$ is a homogeneous polynomial which is positive on the standard simplex, then for sufficiently large $r \in \mathbb{N}$, the polynomial

$$f(x_1, \ldots, x_n) \left( \sum_{i=1}^{n} x_i^2 \right)^r$$

has positive coefficients. Using these two results Parrilo [112] has defined the following cone hierarchy,

$$K^r = \left\{ A \in S_n | P_A(x) \left( \sum_{i=1}^{n} x_i^2 \right)^r \text{ has a sum of square decomposition} \right\}$$

It is shown by Parrilo that the cone $K^r$ approximate $C_n$ from interior since $S_n^+ + N = K^0 \subset K^1 \subset \ldots$ and $int(C_n) \subseteq \bigcup_{r \in \mathbb{N}} K^r$. Hence one can approximate the copositive cone by a set of linear matrix inequality due to the linear matrix inequality formulation of sum of square decomposition. So each copositive program can be approximated by a large SDP. The size of SDP can be exponential in the size of the copositive program.

By definition, $P_A(x)$ has nonnegative coefficients if and only if $A \in N_n$. By exploiting this property de Klerk and Pasechnik [45] has defined the following cone hierarchy,

$$\Lambda^r = \left\{ A \in S_n | P_A(x) \left( \sum_{i=1}^{n} x_i^2 \right)^r \text{ has nonnegative coefficient} \right\}$$

de Klerk and Pasechnik have shown that $N_n = \Lambda^0 \subset \Lambda^1 \subset \ldots$ and $int(C_n) \subseteq \bigcup_{r \in \mathbb{N}} \Lambda^r$. It is not difficult to see that $\Lambda^r \subset K^r$ for all $r = 0, 1, \ldots$ (see [45, Corollary 3.6]), since if $P(x)$ has only nonnegative coefficients then it has sum of square decomposition. The cone program over $\Lambda^r$ result in linear programs since each cone is polyhedral.

The above cone hierarchy for approximating copositive cone results in solving either SDP or linear programs. These programs are exponential in the size of copositive programs. Hence for large $r$ it is difficult to solve these programs. Beside this fact that for large $r$ it takes exponential time to solve certain copositive program, these kind of methods has been applied efficiently for some practical problems (see [24, 44, 45, 110]). To the best of our knowledge no such cone hierarchies exists for the cone of completely positive matrices.
4.3. ALGORITHM

4.3.2 Feasible Descent Method

Bomze et al. [26] has given a feasible descent method for solving optimization problems over the completely positive cones. We will start with following problem,

\[
\begin{align*}
\min & \quad \langle C, X \rangle \\
\text{s.t.} & \quad CPP_1, \quad \langle A_i, X \rangle = b_i, \quad i = 1, ..., m \\
& \quad X \in C_n^*
\end{align*}
\]

Suppose there exists an initial feasible solution \( X^0 \) which has a factorization \( X^0 = (V^0)(V^0)^T \), then the next iteration point can be calculated by \( X^{j+1} = X^j + \nabla X^j \), where \( \nabla X^j \) is a solution of following program,

\[
\begin{align*}
\min & \quad \langle C, \nabla X \rangle + (1 - \epsilon) \|\nabla X\|_j^2 \\
\text{s.t.} & \quad CPP_2, \quad \langle A_i, \nabla X \rangle = 0, \quad i = 1, ..., m \\
& \quad X^j + \nabla X \in C_n^*
\end{align*}
\]

The norm \( \|\cdot\|_j \) used in iteration \( j \) depends on the current iterate \( X^j \). Using \( X^{j+1} = (V + \nabla V)(V + \nabla V)^T \), one can show that the above program can be written as (for details see [26]),

\[
\begin{align*}
\min & \quad \epsilon \langle C, V(\nabla V)^T + \nabla VV^T + (\nabla V)(\nabla V)^T \rangle + \\
& \quad (1 - \epsilon) \|V(\nabla V)^T + \nabla VV^T + (\nabla V)(\nabla V)^T\|_j^2 \\
\text{s.t.} & \quad CPP_3, \quad \langle A_i, V(\nabla V)^T + \nabla VV^T + (\nabla V)(\nabla V)^T \rangle = 0, \quad i = 1, ..., m \\
& \quad V + \nabla V \in N_n
\end{align*}
\]

The objective function of the above program has become a non-convex quadratic function. Algorithm [\textit{\textcircled{1}}] gives an implementable algorithm for solving above regularized problem.

The algorithm [\textit{\textcircled{1}}] gives a solution which is only locally optimal. The convergence of the algorithm is not guaranteed. The main disadvantage of this algorithm is the requirement of an initial solution which itself a hard problem. Moreover finding the factorization of the
4.3. ALGORITHM

**Algorithm 1** Feasible descent method [26, Algorithm 2.1]

**Require:** \( \epsilon \in (0, 1), V \geq O \) with \( \langle A^i, V^T \rangle = b_i \) \( \forall i \)

set \( \nabla V^{old} = O \) and \( \tilde{s}_i = 0 \) for all \( i \). Set \( l = 1 \) and \( \tau_1 = 1 - \epsilon \)

Solve

\[
\min \langle 2C(V + \nabla V^{old}), \nabla V \rangle + (1 - \epsilon) \| \nabla V^{old} + \nabla V \|^2 + \tau_1 \| \nabla V \|^2 \\
\text{s.t.} \\
\langle 2A^i(V + \nabla V^{old}), \nabla V \rangle = \tilde{s}_i, \ i = 1, ..., m \\
V + \nabla V^{old} + \nabla V \in \mathbb{R}^{n \times k}
\]

and denote the optimal solution by \( \nabla V^l \).

Update \( \nabla V^{old} = \nabla V^{old} + \nabla V^l \) and \( \tilde{s}_i = b_i - \langle A^i(V + \nabla V^{old}), V + \nabla V^{old} \rangle \).

If \( \| \nabla V^{old} \| > 1 \) set \( \epsilon = \epsilon/2 \) and \( \nabla V^{old} = \nabla V^{old}/2 \)

If \( \| \nabla V^l \| \approx 0 \) then stop, \( \nabla V^{old} \) approximately solves \( CPP_3 \) locally.

ELSE update \( l = l + 1, \tau_1 = 1.5 \times \tau_{l-1} \), and go to step 2.

The solution matrix is not trivial.

4.3.3 \( \epsilon \)–Approximation Algorithm

An algorithm for the solution of \( CPP \) (see section 4.1) is proposed by [32]. The algorithm is based on some newly defined polyhedral inner and outer approximations of the copositive cones. The idea is similar to the idea of approximating copositive cones by the cone hierarchy. The problem of testing copositivity of a matrix can be converted to an optimization problem over a simplex in following way.

1. A is copositive if and only if \( x^T A x \geq 0 \) for all \( x \in \mathbb{R}_+^n \) with \( \| x \| = 1 \)

2. If we take 1–norm then the set \( \Delta^s = \{ x \in \mathbb{R}_+^n : \| x \|_1 = 1 \} \) is the so called simplex.

3. Hence A is copositive if and only if \( x^T A x \geq 0 \) for all \( x \in \mathbb{R}_+^n \) with \( x \in \Delta^s \)

For describing our algorithm we need to define simplicial partitions,

**Definition 4.15.** Let \( \Delta \) be a simplex in \( \mathbb{R}^n \). A family \( P = \{ \Delta^1, ..., \Delta^m \} \) of simplices satisfying

\[
\Delta = \bigcup_{i=1}^m \Delta^i \ \text{and} \ \text{int}(\Delta^i) \cap \text{int}(\Delta^j) = 0, \forall \ i \neq j
\]

is called a simplicial partition of \( \Delta \).
4.3. ALGORITHM

Moreover the sets of vertices and edges of a simplex are defined as,

\[ V(P) = \{ v : v \text{ is a vertex of some simplex in } P \} \]

\[ E(P) = \{ (u, v) : u \neq v \text{ are vertices of the same simplex in } P \} \]

One can prove that a matrix \( A \) is copositive if \( v^T Av \geq 0 \) for all \( v \in V(p) \) and \( u^T Av \geq 0 \) for all \( (u, v) \in E(P) \) (for details [32, Theorem 3]). Based on the described results we define the two closed convex polyhedral cones \( I_P \) and \( O_P \) giving inner respectively outer approximations of copositive matrix,

\[ I_P = \{ A \in S_n : v^T Av \geq 0, \ \forall v \in V(P) \} \]

\[ u^T Av \geq 0 \ \forall (u, v) \in E(P) \} \]

\[ O_P = \{ A \in S_n : v^T Av \geq 0, \ \forall v \in V(P) \} \]

In the algorithm if the partitions get finer then the approximation of \( C_n \) gets monotonically better for both inner and outer approximation cones. Since the cones \( I_P \) and \( O_P \) are polyhedral, so optimizing over them results in solving linear programming problem. Solving over \( I_P \) gives upper and over \( O_P \) gives lower bounds on optimal solution.

Under standard assumptions the Algorithm 2 is provably convergent. A drawback of this algorithm is that, the constraints in the linear programs grow very quickly and the constraint system contains a lot of redundancy. This results in more computer memory usage. Moreover this algorithm is not applicable for the general model of Burer (see section 4.2.6) and provides poor results for box constrained optimization problems.

Remark Beside the methods/algorithms given above, some other attempts for a solution of copositive programs over copositive or completely positive matrices are given in [17, 22, 35].
Algorithm 2 $\epsilon$-Approximation Algorithm for $CP_P$

1: set $P = \{\Delta^s\}$
2: solve the inner $LP$

$$\min\langle C, X \rangle$$
$$\text{s.t.}$$
$$\langle A_i, X \rangle = b_i, \ i = 1, ..., m$$
$$X \in I_P$$

let $X^I$ denotes the solution of this program

3: solve the outer $LP$

$$\min\langle C, X \rangle$$
$$\text{s.t.}$$
$$\langle A_i, X \rangle = b_i, \ i = 1, ..., m$$
$$X \in O_P$$

let $X^O$ denotes the solution of this program

4: if $\frac{\langle C, X^I \rangle - \langle C, X^O \rangle}{1 + ||\langle C, X^I \rangle|| + ||\langle C, X^O \rangle||}$ then
5: STOP $X^O$ is an $\epsilon-$ optimal solution of $CP_P$
6: end if

7: choose $\Delta \in P$
8: bisect $\Delta = \Delta^1 \cup \Delta^2$
9: set $P \leftarrow P\{\Delta\} \cup \{\Delta^1, \Delta^2\}$
Chapter 5

Semi-Infinite Programming (SIP) 
Representation of CP and SDP

SIP, CP and SDP being subclasses of mathematical programming have some obvious connections. In special cases these three classes can be represented by each other. In previous chapters we have discussed about CP and SDP. In this chapter we will investigate the connections among SIP, CP and SDP. We will start this chapter by giving description of semi-infinite programming. Then we will give SIP representation of CP and SDP. The final section of this chapter is devoted to SIP solution methods for SDP.

5.1 Semi-infinite Programming (SIP)

Semi-infinite programming is one of the oldest classes of mathematical programming. Being oldest class it is very well studied in literature. The area of semi-infinite programming is very rich in terms of theoretical results. A major drawback of semi-infinite programming is the nonexistence of an efficient algorithm which can solve semi-infinite programs to optimality. In special cases some other classes like cone programs can be represented as semi-infinite programs.

The most general form of SIP is,

$$\min_{x \in \mathbb{R}^n} f(x)$$

s.t.

$$P_{SIP} \quad g(x, v) \geq 0 \quad \forall \ v \in V$$
where $f, g$ are real valued (continuous) functions and $V$ is a compact set. The set of feasible points for $P_{SIP}$ is defined by infinitely many constraints,

$$Feas(P_{SIP}) = \{x | g(x, v) \geq 0, v \in V\}$$

It is not difficult to show that $Feas(P_{SIP})$ is closed. Moreover $P_{SIP}$ will be convex if the function $f$ is convex and for every index $v \in V$, the constraint function $g_v(x) = g(x, v)$ is concave that is $-g_v$ is convex. In convex SIP the set of feasible points is always convex and every local minimizer is also global minimizer. First and second order optimality conditions have been derived for Semi-infinite programming. These conditions are based on generalized Farkas lemma and generalized Mangasarian-Fromowitz constraint qualification.

SIP arise naturally in many applications for example Chebyshev approximation. SIP has been applied to some other areas like minimal norm problem in the space of polynomial, mathematical physics, robotics, geometry, optimization under uncertainty, economics, robust optimization and system and control (see [79, 101, 139]).

Since the feasibility problem for SIP cannot be solved in polynomial time in general hence existence of polynomial time algorithm for SIP is not possible unless $P = NP$. Nonexistence of an efficient algorithm implies that one has to rely on numerical methods. The numerical methods available can be classified to three main categories: Local reduction method, discretization methods and exchange methods.

In the local reduction method the original problem is replaced by a locally equivalent problem with finitely many inequality constraints. The problem can also be replaced by a system of nonlinear equations with finitely many unknowns. This system can be solved by Newton’s method and hence these methods have good convergence results. Reduction based SQP-methods are one example of these kind of methods.

Discretization methods are based on solving a sequence of finite programs. The sequence of finite programs are solved according to some pre-defined grid generation scheme or some cutting plane scheme.

The exchange methods can be seen as a compromise between discretization methods and reduction methods. Hence they are more efficient then discretization methods. Exchange methods works in two phases, for complete details see [78, 79, 122].

Besides the methods discussed above, some other methods also exists for solution/approximation of SIP. These methods includes interior point method [134, 136] and feasible direction methods [101, 122].
5.1. SEMI-INFINITE PROGRAMMING (SIP)

Remark The earliest description of SIP can be found in 1924, where it was discussed as Chebyshev approximation. The term semi-infinite was first used by Charnes and Kortanek in some papers devoted to linear semi-infinite programs (LSIP) (see [101]). Now the field of semi-infinite programming is rather very well developed with respect to both theoretical as well as practical point of view. In last decades more then 1000 articles and 10 books have been published covering one or more aspect of SIP (see the huge list of references on SIP collected by Lopez and Still [100]).

5.1.1 Linear Semi-infinite Program (LSIP)

In SIP when all the constraints and objective functions are linear then we will have a linear semi-infinite program (LSIP). One can write LSIP in the following form,

$$
\begin{align*}
\min_{x \in \mathbb{R}^n} & \quad c^T x \\
\text{s.t.} & \quad a_v^T x \geq b_v \quad \forall \quad v \in V
\end{align*}
$$

where $a_v, c \in \mathbb{R}^n$ and $b_v \in \mathbb{R}$. The set of feasible points of $P_{LSIP}$ is a closed convex set, since it is an intersection of closed half spaces. One can define different dual problems associated with $P_{LSIP}$. If $P_{LSIP}$ is continuous i.e. when $V$ is compact and $a_v, b_v$ are continuous function on $V$, then one can define the following dual,

$$
\begin{align*}
\max \int_V b(v)d(u(v)) \\
\text{s.t.} & \quad \int_V b(v)d(u(v)) = c \\
& \quad u \in K^+(V)
\end{align*}
$$

where $v \mapsto a_v \equiv a(v)$ and $v \mapsto b_v \equiv b(v)$ and $K^+$ is the cone of non-negative Borel measures on the compact space $V$. Another dual which can be associated with $P_{LSIP}$ is the so called
5.1. SEMI-INFINITE PROGRAMMING (SIP)

Haar dual defined below,

\[
\max \sum_{v \in V} u_v b_v \\
\text{s.t.} \\
D_{LSIP} \sum_{v \in V} u_v a_v = c \\
u_v \geq 0, \quad v \in V
\]

where only a finite number of dual variables \( u_v \), \( v \in V \) attain positive values. \( D_{LSIP} \) is feasible if and only if \( c \) belongs to the cone generated by \( a_v \), that is

\[
D_{LSIP} \text{ is feasible if and only if } c \in \text{cone}\{a_v|v \in V\} \tag{5.1}
\]

Remark It is worth mentioning that the sum \( \sum_{v \in V} u_v b_v \) is finite, since by Caratheodory’s theorem (see e.g. [55, Theorem 3.6]) this sum can be express with at most \( n \) non-zero coefficients \( u_y \). The same is true for the sum \( \sum_{v \in V} u_v a_v \).

Let us write \( u = (u_v) \) then,

**Lemma 5.1.** If \( x \in \text{Feas}(P_{LSIP}) \) and \( u \in \text{Feas}_{D_{LSIP}} \) then \( c^T x \geq \sum_{v \in V} u_v b_v \)

**Proof.** see e.g. [55, Theorem 12.17]. \( \square \)

Before giving the strong duality result we give some basic definitions required for the strong duality proof.

**Definition 5.2 (Active Index Point).** Let the vector \( \bar{x} \in \text{Feas}(LSIP) \) then the active index set denoted by \( V(\bar{x}) \) is given below,

\[
V(\bar{x}) = \{\bar{v} \in V|g(\bar{x}, \bar{v}) = 0\} \tag{5.2}
\]

The set \( V(\bar{x}) \) is a closed compact subset of \( V \).

A strictly feasible direction at \( \bar{x} \) is a vector \( d \in \mathbb{R}^n \) such that,

\[
\nabla x g(\bar{x}, \bar{v}) d < 0 \quad \forall \ \bar{v} \in V(\bar{x})
\]
5.1. SEMI-INFINITE PROGRAMMING (SIP)

Definition 5.3 (KKT Condition). A local minimizer \( \pi \) is said to satisfy the KKT condition if there exist multipliers \( \mu_1, ..., \mu_k \geq 0 \) and indexes \( \bar{v}_1, ..., \bar{v}_k \in V(\pi) \) such that,

\[
\nabla f(\pi) + \sum_{j=1}^{k} \mu_j \nabla_x g(\pi, \bar{v}_j) = 0^T
\]

(5.3)

Just like cone programming strong duality does not hold in general for \( LSIP \), we need to define a constraint qualification.

Definition 5.4 (Slater Condition). The primal Slater condition holds if there exists \( x \in \mathbb{R}^n \) with \( a_v x > b_v \) \( \forall v \in V \)

(5.4)

We say dual Slater condition holds if

\[
c \in \text{int}(\text{cone}\{a_v|v \in V\})
\]

(5.5)

Now we will state and prove strong duality result.

Theorem 5.5. \footnote{This theorem is extracted from \cite[Theorem 12.18,12.19 and corally 12.3]{55}} If the primal (5.4) and dual slater (5.5) conditions are satisfied. Then primal and dual solutions \( x \) and \( u \) exists and \( \text{val}(P_{LSIP}) = \text{val}(D_{LSIP}) \).

Proof. Let \( A \) is a matrix with infinite mainly rows \( a_v^T, v \in V \) and \( b \) is a vector with infinite components \( b_v \). The dual slater condition (5.5) implies that \( D_{LSIP} \) is feasible. Hence weak duality implies that \( P_{LSIP} \) is bounded.

On the basis of the dual Slater condition we may assume that there exists a strictly feasible dual basic solution,

\[
\hat{y}^T = c^T \hat{A}^{-1} > 0^T
\]

where \( \hat{A} \) is a basic submatrix of \( A \). Scaling \( c \) if necessary we may further assume \( 0 < \hat{y} < 1 \).

Let \( \epsilon > 0 \) such that for all \( \tilde{c} = c + \epsilon p, \|p\| \leq 1 \),

\[
\tilde{y}^T = \tilde{c}^T \hat{A}^{-1} \quad \text{still satisfies} \quad 0 \leq \tilde{y} \leq 1
\]

(5.6)

Let \((x_k)\) be a sequence of primal feasible solution such that \( c^T x_k \to \text{val}(P_{LSIP}) \) and take \( p = \frac{x_k}{\|x_k\|} \), with the assumption that for sufficiently large \( k, x_k \neq 0 \). Hence we have perturbation
5.1. SEMI-INFINITE PROGRAMMING (SIP)

\[ \tilde{c} = c + \epsilon \frac{x_k}{\|x_k\|} \] Since we have \( c^T x_k \to val(P_{LSIP}) \), hence for sufficiently large \( k \),

\[
val(P_{LSIP}) - \epsilon \leq c^T x_k = \tilde{c}^T - \epsilon \|x_k\| \\
= \bar{y}^T_k \hat{A}x_k - \epsilon \|x_k\| \\
\leq \bar{y}^T_k \hat{b} - \epsilon \|x_k\|
\]

where \( \hat{b} \) is the right hand side subvector of \( b \) corresponding to \( \hat{A} \). Hence,

\[ \epsilon \|x_k\| \leq \bar{y}^T_k \hat{b} - valP_{LSIP} + \epsilon \leq \sum_{j=1}^{n} \|\hat{b}_j\| - val(P_{LSIP}) + \epsilon \]

shows that \( \|x_k\| \) is bounded. So \( (x_k) \) must contain a convergent subsequence. Giving that \( \bar{x} = \lim x_s \), hence \( \bar{x} \) is feasible since \( Feas(P_{LSIP}) \) is closed set and \( val(P_{LSIP}) = \bar{x} \). This proofs the existence of an optimal solution \( \bar{x} \).

Now assume that \( \hat{x} \) is a Slater point for \( P_{LSIP} \). Then take \( d = \hat{x} - \bar{x} \), then for any \( v \in V(\bar{x}) \), we will have,

\[
\nabla_x g(\bar{x}, v)d = a_v^T d = a_v^T (\hat{x} - \bar{x}) < b_v - b_v = 0
\]

giving that \( d \) is a feasible direction. Since we have discovered a feasible direction \( d \) at \( \bar{x} \) hence \( \bar{x} \) satisfies KKT-condition,

\[ 0^T = \nabla f(\bar{x}) + \sum_{j=1}^{n} \mu_j \nabla x g_j(\bar{x}, v_j) = -c^T + \sum_{j=1}^{n} \mu_j a_{v_j}^T \]

with \( \mu_j \geq 0, v_j \in V(\bar{x}) \). So \( \bar{y}_{v_j} = \mu_j \) proves the claim. Now by definition of \( \bar{y}_v \), we find \( \bar{y}^T (b - A(\bar{x})) = \sum_j \bar{y}_{v_j} (b_{v_j} - a_{v_j}^T \bar{x}) = 0 \) hence \( \bar{y}^T b = c^T \bar{x} \) implying \( val(P_{LSIP}) = valD_{LSIP} \).

From (5.1) we get the intuition that there is some relationship between LSIP and cone programming. Our next two sections are focused on this kind of relations.

**Remark** Although LSIP is linear and convex but the existence of polynomial time algorithm is not possible for LSIP. The main difficulty lies in checking the constraint \( g(x, v) \geq 0 \ \forall v \in V \). Checking the feasibility of point \( \bar{x} \) will result in solving a global minimization
5.2 SIP REPRESENTATION OF CP

Consider the following copositive program,

$$\min_{y} \ g(\bar{x}, v)$$

s.t.

$$v \in V$$

and to check that for a global minimum $\tau$ the condition holds $g(\bar{x}, \tau) \geq 0$.

5.2 SIP Representation of CP

Consider the following copositive program,

$$\min_{x \in \mathbb{R}^m} \ c^T x$$

s.t.

$$P_{CP} \quad \sum_{i=1}^{m} x_i A_i + Z = B$$

$$Z \in C_n$$

where $A_i, B \in S_n$. Recall from Definition 2.13 that for any $A \in S_n$, we will have,

$$A \in C_n \iff v^T A v \geq 0 \ \forall \ v \in V = \{v \in \mathbb{R}^n_+ \mid \|v\| = 1\}$$

By the above definition we can have,

$$Z \in C_n \Rightarrow B - \sum_{i=1}^{m} x_i A_i \in C_n$$

$$\Rightarrow v^T \left( B - \sum_{i=1}^{m} x_i A_i \right) v \geq 0 \ \forall \ v \in V$$

$$\Rightarrow a_v^T x - b_v \geq 0 \ \forall \ v \in V$$
5.2. SIP REPRESENTATION OF CP

where \(a^T_v = (v^T A_1 v, v^T A_2 v, ..., v^T A_m v)\) and \(b_v = v^T B v\). Then we can write \(P_{CP}\) equivalently as,

\[
\min_{x \in \mathbb{R}^m} c^T x \\
\text{s.t.} \\
P_{CP} \quad a^T_v x - b_v \geq 0 \quad \forall v \in V
\]

Above is a LSIP with constraints \(a^T_v x \geq b_v\), hence its dual can be written as

\[
\max_y \sum_{v \in V} b_v y_v \\
\text{s.t.} \\
(D_{CP}) \quad \sum_{v \in V} y_v a_v = c \\
y_v \geq 0 \quad \forall v \in V
\]

Now if we consider,

\[
v^T B v - \sum_{k=1}^m x_k v^T A_k v = \sum_{i=1,j=1}^m b_{ij} v_i v_j - \sum_{k=1}^m x_k \sum_{i=1,j=1}^m (a_{ij}) k v_i v_j \\
= \langle vv^T, B \rangle - \sum_{k=1}^m x_k \langle vv^T, A_k \rangle
\]

then \((P_{CP})\) and \((D_{CP})\) can be written as follows,

\[
\min_{x \in \mathbb{R}^m} c^T x \quad \text{max}_y \sum_{v} y_v \langle vv^T, B \rangle \\
\text{s.t.} \quad \sum_{k=1}^m x_k \langle vv^T, A_k \rangle \leq \langle vv^T, B \rangle \quad \forall v \in V \\
P_{CP} \quad s.t. \quad \sum_{v} y_v \langle vv^T, A_k \rangle = c_k, k = 1, ..., n \\
D_{CP} \quad y \geq 0
\]
5.2. SIP REPRESENTATION OF CP

Considering the fact that $C_n^*$ can also be described as $C_n^* = \text{cone}\{vv^T|v \in \mathbb{R}_+^n\}$ we can use substitution, $Y = \sum y_v[vv^T] \in C_n^*$, hence primal and dual can take the form,

$$\begin{align*}
\min_{x \in \mathbb{R}^n} & \quad c^T x \\
\text{s.t.} & \quad \sum_{i=1}^m x_i A_i + Z = B \\
\text{P}_{CP} & \\
\max_Y & \quad \langle Y, B \rangle \\
\text{s.t.} & \quad \langle Y, A_k \rangle = c_k, \quad k = 1, ..., m \\
\text{D}_{CP} & \\
Z & \in C_n \\
Y & \in C_n^*
\end{align*}$$

**Theorem 5.6 (Weak Duality).** Let $x$ and $Y$ belongs to the set of feasible points of $P_{CP}$ and $D_{CP}$ respectively then, $c^T x \leq \langle Y, B \rangle$.

**Proof.** Similar to the proof of Lemma 3.1.

Recall, for strong duality result the characterization of interior for primal and dual programs is often required. It is well known that the interior of copositive cone consist of the so called strictly positive cones, i.e.

$$\text{int}(C_n) = \{ A \in S_n | x^T A x > 0, x \in \mathbb{R}_+^n \}$$

where $S_n$ is the set of symmetric matrices, hence the primal slater condition will be of the form,

$$\exists \; \hat{x} \in \mathbb{R}^n \; \text{s.t.} \; Z = \left( B - \sum_{i=1}^m \hat{x}_i A_i \right) \in \text{int}(C_n) \quad (5.7)$$

Since $a_v = (\langle A_1, [vv^T] \rangle, ..., \langle A_m, [vv^T] \rangle)$ and the definition of completely positive cone, we can write,

$$c \in \text{int}\{(\langle A_1, Y \rangle, ..., \langle A_m, Y \rangle)^T | Y \in C_n^* \} \quad (5.8)$$

Hence we will state the strong duality result,

**Theorem 5.7.** If the constraints qualification (5.7) and (5.8) holds then both primal and dual have complementary optimal solution $(x)$ respectively $(Y)$.

**Proof.** The proof is similar to the proof of Theorem 5.5.
5.3 SIP Representation of SDP

Consider the SDP with linear constraints and linear objective function,

\[
\min_{x \in \mathbb{R}^m} \quad c^T x \\
\text{s.t.} \\
P_{SDP} \quad \sum_{i=1}^{m} x_i A_i + Z = B \\
Z \succ 0
\]

where \( A_i, B \) are \( n \times n \) symmetric matrices. We know that for any \( A \in S^+_n \), we will have,

\[
A \succ 0 \iff v^T A v \geq 0 \quad \forall \ v \in V = \{ v \in \mathbb{R}^n_+ \mid \|v\| = 1 \}
\]

by above definition we can have,

\[
Z \succ 0 \Rightarrow v^T \left( B - \sum_{i=1}^{m} x_i A_i \right) v \geq 0 \quad \forall \ v \in V \\
\Rightarrow a_v^T x - b_v \geq 0 \quad \forall \ v \in V
\]

where \( a_v^T = (v^T A_1 v, v^T A_2 v, ..., v^T A_m v) \) and \( b_v = v^T B v \). Then we can write \( P \) equivalently as,

\[
\min_{x \in \mathbb{R}^m} \quad c^T x \\
\text{s.t.} \\
P_{SDP} \quad a_v^T x - b_v \geq 0 \quad \forall \ v \in V
\]
5.3. **SIP REPRESENTATION OF SDP**

Above is LSIP with constraints $a_v^T x \geq b_v$, hence its dual can be written as

$$\min_y \sum_{v \in V} b_v y_v$$

**s.t.**

$$D_{SDP} \sum_{v \in V} y_v a_v = c$$

$$y_v \geq 0 \ \forall v \in V$$

$P_{SDP}$ and $D_{SDP}$ can also be written as follows,

$$\min_{x \in \mathbb{R}^m} c^T x$$

**s.t.**

$$P_{SDP} \sum_{k=1}^{m} x_k \langle vv^T, A_k \rangle \leq \langle vv^T, B \rangle \ \forall v \in V$$

$$D_{SDP} \sum_{v} y_v \langle vv^T, A_k \rangle = c_k, \ k = 1, ..., n$$

$$y \geq 0$$

Using the substitution $Y = \sum y_v [vv^T] \in (S_n^+)^* = S_n^+$, $D_{SDP}$ can be written as,

$$\max_Y \langle Y, B \rangle$$

**s.t.**

$$D_{SDP} \langle Y, A_k \rangle = c_k, \ k = 1, ..., m$$

$$Y \in S_n^+$$

By Using above $P_{SDP}$ and $D_{SDP}$ weak and strong duality theorem can be formulated in a similar manner as was done in last section. We will use LSIP representation of SDP in next section.
### 5.4 SIP Solution Approach for SDP

The primal and dual pair of SDP program can be written as,

\[
\begin{align*}
\text{max} \quad & \langle Y, B \rangle \\
\text{s.t.} \quad & P \langle A_i, Y \rangle = c_i, \quad \forall i = 1, \ldots, m \\
\text{min} \quad & c^T x \\
\text{s.t.} \quad & D g(x) = A_0 - \sum_{i=1}^{m} A_i x_i \\
\end{align*}
\]

where \( A_i \in S_n \quad \forall i = 1, \ldots, m \) and \( C \) are given matrices and \( c = \{c_1, c_2, \ldots, c_m\} \in \mathbb{R}^m \). A matrix is positive definite if all its eigenvalues are positive. So the dual program can also be written as follows,

\[
\begin{align*}
\text{min} \quad & c^T x \\
\text{s.t.} \quad & EOP \lambda_j(x) \geq 0, \quad \forall j = 1, \ldots, n
\end{align*}
\]

where \( \lambda_j(x) \) is the function of eigenvalues of the matrix \( g(x) \). Now if we consider \( g(x) \) as parametric family of matrices depending linearly on parameter \( x \), then we can write the eigenvalue equation for \( g(x) = A_0 - \sum_{i=1}^{m} A_i x_i \) as,

\[
g(x)v = \lambda(x)v
\]

Now consider \( \lambda_j : \mathbb{R}^n \to \mathbb{R} \) as function of eigenvalues and \( v_j : \mathbb{R}^n \to \mathbb{R}^n \) as the corresponding set of orthonormal eigenvectors for the above equation. One can see \( D \) and \( EOP \) are same problem. But there are two major differences in \( D \) and \( EOP \),

1. The constraint in \( D \) are convex while in \( EOP \) they are concave,

2. In \( D \) constraint are differentiable. For \( EOP \) its is well known that the function \( \lambda_j \) is continuous but differentiability is not satisfied.

The reason why we have converted \( SDP \) to \( EOP \) is the unavailability of fast solution method for \( SDP \). Although interior point methods are polynomial time methods, but for most of practical problems where we have thousands of constraints these methods become
5.4. SIP SOLUTION APPROACH FOR SDP

annoying.

Here we will try to solve SDP problems by exploiting the structure of eigenvalues. So we begin by investigating some properties of the eigenvalue function. Let us start with the simple case when all the eigenvalues of the $g(x)$ are simple for a fixed $\bar{x} \in \mathbb{R}^n$, then we will have following result,

**Theorem 5.8.** Suppose that $g(x) \in C^r(\mathbb{R}^n, S_n)$ for some $r \geq 0$. Let $\lambda : \mathbb{R}^n \rightarrow \mathbb{R}$ be a fixed eigenvalue function of $g(x)$ with corresponding eigenvector function $v : \mathbb{R}^n \rightarrow \mathbb{R}^n$. If $\lambda(\bar{x})$ is a simple eigenvalue of $g(\bar{x})$ locally near $\bar{x}$ then $\lambda \in C^r(\mathbb{R}^n, \mathbb{R})$ and $v \in C^r(\mathbb{R}^n, \mathbb{R}^n)$.

**Proof.** For the proof we consider the following function,

$$H(x, \lambda, v) = \begin{cases} g(x)v - \lambda v = 0 \\ v^Tv - 1 = 0 \end{cases}$$  \hspace{1cm} (5.9)

It can be easily verified that for a fixed $\bar{x} \in \mathbb{R}^n$, if $(\lambda, v)$ is a solution of $H(\bar{x}, \lambda, v) = 0$ then $(\lambda, v)$ is an eigenpair of $g(x)$. Now consider $H \in C^r(\mathbb{R}^n, \mathbb{R})$ since $g(x) \in C^r(\mathbb{R}^n, S_n)$. Moreover, the Jacobian $|D_{\lambda,v}H(\bar{x}, \lambda, v)| \neq 0$, where $|A|$ denotes determinant of $A$. To prove this, we consider the Jacobian matrix $J = D_{\lambda,v}H(\bar{x}, \lambda, v)$ of (5.9),

$$J = \begin{bmatrix} g(\bar{x}) - \lambda I - v \\ 2v \\ -\bar{v} \end{bmatrix}$$

$J$ is singular if for some $a \in \mathbb{R}^n$ and $\beta \in \mathbb{R}$, the system,

$$(g(\bar{x}) - \lambda I)a - \beta \bar{v} = 0$$

$$2\bar{v}^Ta = 0$$

has only trivial solution, i.e. $a = 0, \beta = 0$. Multiplying the first equation by $\bar{v}^T$ from left we will get $\beta = 0$, hence our system reduces to,

$$(g(\bar{x}) - \lambda I)a = 0$$

$$2\bar{v}a = 0$$

First equation says that $a$ belongs to the null space of $g(\bar{x}) - \lambda I$. So we can write $a = \alpha \bar{v}$.
5.4. SIP SOLUTION APPROACH FOR SDP

So from second equation we will get

\[ 2\alpha \bar{v}^T \bar{v} = 0 \]

and by \( \bar{v}^T \bar{v} = \| \bar{v} \|^2 \neq 0 \) we find \( \alpha = 0 \), hence \( a \) is a zero vector. So \( |D_{\lambda,v} H(\bar{x}, \bar{v}, \bar{v})| \neq 0 \), now by implicit function theorem locally near \( \bar{x} \), \( \lambda \in C^r(\mathbb{R}^n, \mathbb{R}) \) and \( v \in C^r(\mathbb{R}^n, \mathbb{R}^n) \).

It is interesting to note that in case of eigenvector function, the continuity of the eigenvector function is not guaranteed even in one parameter case, see the example below.

**Example** Consider the one parameter matrix, depending on \( x \in \mathbb{R} \),

\[ g(x) = e^{-\frac{1}{x^2}} \begin{pmatrix} \cos\frac{2}{x} & \sin\frac{2}{x} \\ \sin\frac{2}{x} & -\cos\frac{2}{x} \end{pmatrix} \]

\( g(x) \) is infinitely differentiable on \( \mathbb{R} \). The eigenvalue function is

\[ \lambda(x) = \begin{cases} \pm e^{-\frac{1}{x^2}} & x \neq 0 \\ 0 & x = 0 \end{cases} \]

But we can not define any eigenfunction which is continuous at \( x = 0 \).

If the eigenvalue \( \lambda(x) \) is not simple we do not have smoothness result for eigenvectors but these results holds for eigenvalues.,

**Theorem 5.9.** Assume \( g \in C^r(\mathbb{R}, S_n), r \in \mathbb{N} \cup \{\infty\} \). The the eigenvalue function \( \lambda_j \) of \( F \) can be defined in such a way that \( \lambda_j \in C^r(\mathbb{R}^n, \mathbb{R}), j = 1, \ldots, n \)

Coming back again to \( EOP \) and \( D \). Owing to the SIP representation of SDP we can write,

\[ g(x) \succeq 0 \iff v^T g(x)v \geq 0 \forall v \in V \]

With the assumption that Slater condition holds for \( P \), the minimum of \( EOP \) can only be obtained at minimum eigenvalue identified by the active index set since for \( v \in V(\bar{x}) \) we
5.4. SIP SOLUTION APPROACH FOR SDP

have \( v^T g(x)v = 0 \), so \( D \) can be written as,

\[
\begin{align*}
\min & \quad c^T x \\
\text{s.t.} & \quad \lambda_{\min}(v^T g(x)v) = 0 \quad \forall \quad v \in V(\bar{\pi})
\end{align*}
\]

**Remark** In [29], Bossum has discussed Newton type methods based on formulation of SDP as eigenvalue problem. Krishnan and Mitchel [90] has developed methods based on SIP representation of SDP and spectral bundle methods. With the help of example problems like bisection and MaxCut they have compared their method with interior point methods.
Chapter 6

Conclusion and Future Work

The main objective of this thesis was to explore the relationship between three different classes of mathematical programming. Throughout this thesis we have concentrated on copositive programming and related concepts. We have dealt with copositive, semidefinite and semi-infinite programming with respect to integer programming or to be more precise with respect to combinatorial optimization. In this chapter we will conclude about our findings in this thesis. Since our emphasis is survey existing results so only few new results are included in this thesis. In the last section of this chapter we will enlist some future work associated with the topic of this thesis or more precisely with copositive programming.

6.1 Conclusion

Here we will discuss about our findings and results we have discussed in each chapter. If some new results are included in any of chapter here we will pinpoint those results.

First chapter is introductory. The literature review part contains references to almost all state of the art results for copositive programming. We have also given some literature on semidefinite programming. The literature indicated in chapter 1 can be useful for new researchers working in the field of copositive programming.

In the second chapter we have tried to collect some state of the art results on the cone of completely positive and copositive matrices which we feel necessary for their application in copositive programming. Our contribution in this chapter is Lemma 2.21. To the best of our knowledge no similar result is available in literature. Moreover we have surveyed some recent developments for the cone of completely positive and copositive matrices.

Chapter 3 is mainly a revision of some old results known for cone programming and
semidefinite programming. Our main contribution in this chapter is an easy proof of results by Kojima and Tuncel \[88\] for quadratically constrained quadratic programs and its relaxations. Other contribution includes the indication of literature for different problem for which semidefinite programming is proved useful.

Chapter 4 is our main chapter. In this chapter we have mainly dealt with the survey of applications and methods/algorithm available for copositive programming. Bundfuss \[30\] has also given the survey of applications but we have attempted to provide surveys on available algorithm/solution procedure for copositive programming. Our main contribution in this chapter is subsection [4.2.2]. We have extended the result of Kojima and Tuncel (see \[88\] or subsection [3.2.2]) for the case of copositive programming. The result can be seen as a positive step for determining copositive relaxation for quadratically constrained quadratic programs.

In chapter 5 we have briefly described semi-infinite programming. It is well known that linear conic programs can be represented as linear semi-infinite programs but no results are available for copositive programming. We have represented copositive program as linear semi-infinite program with the future objective to apply linear semi-infinite program methods for solving/approximating copositive programs. Moreover semi-infinite representations of copositive programming is useful for developing duality results for copositive programming. In the last section of the chapter we have represented semidefinite programming as eigenvalue problem and proved some associated results for eigenvector and eigenvalue function. It is well known fact that semidefinite programming problems can be solved by the methods of eigenvalue optimization. Spectral bundle methods are well known in this regard. We have formulated the semidefinite programing problem as eigenvalue optimization problem with the future goal of obtaining similar results for copositive programming.

6.2 Future Work

During the literature search and compiling of the thesis we have kept an eye on open problems related to the topic of the thesis which can be interesting for future work. Here we will list some of them with brief detail.

1. **Copositive Programming and Burer’s MBQP** (see subsection [4.2.6])
   - Extension of Burer’s result to the case of quadratic constraints.
   - The dual of Burer’s completely positive representation of \(QP\) is a copositive
6.2. FUTURE WORK

program. It is still not known whether there is any duality gap in copositive and completely positive representation. This will be an interesting problem since there exists some approximation methods for programs over the cone of copositive matrices but no such methods exists for programs over the cone of completely positive matrices.

2. Copositive Programming and Semi-infinite Programming

- Investigating the use of methods available for semi-infinite programming for solving copositive programs.
- Since copositive programs can be represented as semi-infinite programs so the use of discretization methods might result in good solution methods for copositive programming. Cone hierarchy based methods can be seen as discretization methods. Hence we can expect good solution methods.
- Semi-infinite representation of semidefinite programs allows to deploy methods of eigenvalue optimization for solving semidefinite programs. The similar can be investigated for the case of copositive programming.

3. Copositive and Completely Positive Cones

- The complexity of completely positive matrices is conjectured (infact strongly believed) to be NP-hard but no formal proof is given, it would be interesting to give a formal proof.
- By further exploring the geometrical aspects of copositive matrices such as faces of the copositive cone can help to develop more efficient algorithms.
- Finding characterization for extremal rays for copositive matrices.
- It would be nice to find some necessary and sufficient conditions for checking copositivity or complete positivity for certain classes of matrices.
- It is already known that a binary symmetric matrix is completely positive if and only if it is semidefinite. The question is "Is it easy to factor a binary matrix into another binary matrix". For the three dimensional case Berman and Xu [14] has formulated necessary and sufficient conditions for matrices with integer entries but nothing is known for higher dimension.
- The so called CP-rank of completely positive matrices is conjectured to be $\frac{n^2}{4}$. This conjecture is already proved for the special class of 5 dimensional matrices.
containing at least one zero. The proof for general class of matrices would be interesting.

4. Miscellaneous

- Extension of PTAS given by Bomze and De Klerk [24], to the more general class of quadratic programs.
- Finding cone hierarchy for completely positive matrices.
- Semidefinite relaxation for MaxCut and interior point methods resulted in 0.87 randomized approximation algorithm. It would be interesting to investigate if we can get a similar result using copositive programming.
- Finding special cases when we can have equality in Theorem 4.3
- Formulation of duality results which are specific for copositive programs can be a big achievement for the development of new solution algorithms.
References


REFERENCES


REFERENCES


REFERENCES


REFERENCES


REFERENCES


REFERENCES


REFERENCES


REFERENCES


REFERENCES


REFERENCES


REFERENCES


REFERENCES


REFERENCES


