On the depth of the Branch and Bound tree for solving the Maximum Stable Set Problem

Inaugural dissertation

for the attainment of the title of doctor
in the Faculty of Mathematics and Natural Sciences
at the Heinrich Heine University Düsseldorf

presented by

Fatemeh Baniasadirad
from Kerman, Iran

Düsseldorf, February 2018

from the institute for Mathematics
at the Heinrich Heine University Düsseldorf

Published by permission of the
Faculty of Mathematics and Natural Sciences at
Heinrich Heine University Düsseldorf

Supervisor: Prof. Dr. Florian Jarre
Co-supervisor: Prof. Dr. Achim Schädle

Date of the oral examination: 01.06.2017

Abstract

Motivated by the recent progress in the solution of large scale semidefinite programs, in particular with the program package SDPNAL [35], this thesis investigates in how far semidefinite programs can be used in a branch and bound approach for solving the maximum stable set problem.
Chapter 2 provides an introduction to semidefinite programs.
Chapter 3 briefly describes how SDPNAL solves semidefinite programs. Lemma 7, the proof of Lemma 8, Remark 2 and Example 2 are the main contributions of this chapter.
The maximum stable set problem and some known facts are presented in Chapter 4. Classical relaxations for the maximum stable set problem are the Lovász θ-number (4.8) and the Lovász-Schrijver θ'-number (4.9). We recall some well known alternative relaxations. In Theorem 10, based on the proof of M. Laurent [20], the proof of equivalency of two such formulations (4.6) and (4.8) is extended to the doubly nonnegative cone. Theorem 9 presents a new proof for the relation of (4.6) with another relaxation (4.7). A new relaxation of (4.7) is introduced in problem (4.18). According to Proposition 3, all semidefinite relaxations in this chapter are mathematically equivalent. A numerical comparison of them given in Table 4.1 is intended to illustrate possible rounding errors associated with the approximate solution generated by SDPNAL. The last discussion of this chapter is concerned with the computation a valid upper bound on the optimal value of a primal semidefinite program from an approximate solution under generic nondegeneracy conditions.
The last four chapters present and analyze the branch and bound strategy to solve the maximum stable set problem. At each level of branching a series of semidefinite relaxations is solved providing a valid upper bound to the stable set problem. We complement the semidefinite upper bound by modifying a randomized approach (due to Goemans and Williamson) to generate feasible and locally optimal solutions to the stable set problem. We show (Proposition 1) that the distribution for generating the approximate solution to the maximum stable set problem is not dependent on the factorization of the optimal solution of the semidefinite relaxation. The main contributions of these chapters are listed as follows:

1. To avoid inaccurate results caused by rounding errors and truncation errors in the solution generated by SDPNAL, we propose a practical method applied to an approximate solution of the Lovász-Schrijver number (4.9). The method provides a reliable upper bound on the stable set problem.

2. We develop a new approach for estimating the stability number of a given graph (Section 6.1). To apply this method, we assume that an upper bound on the stable set problem is known. We introduce a random $\{0,1\}$-vector and solve a number of relaxations (4.6) for a graph defined via the $\{0,1\}$-vector. Three different strategies are presented to construct a random $\{0,1\}$-vector with the goal of reducing the dimension of the relaxations (4.6)

(Section 6.1.1).

3. Another approach contributed in this thesis is adding additional constraints to the relaxation (4.6) to strengthen the upper bound on the stable set problem and to provide a new branching strategy. We discuss how to determine the data of the new additional constraint to provide a deep cut (Section 6.3). We also address how to prove the infeasibility of the new relaxation in the presence of rounding errors (Section 6.4).

4. We discuss the relation between the algorithm proposed in this dissertation and the one proposed by Benson and Ye through Corollary 1 and Remark 14. Lemma 12 (not proven by Benson and Ye) determines the Lovász-number of the Benson-Ye-approach. The last chapter gives some insight about the practical behavior of a branch and bound approach.

Zusammenfassung

Motiviert durch die neuesten Entwicklungen bei der Lösung von “ large scale ” Semidefiniten Programmen, insbesondere mit den Programmpaket SDPNAL [35], untersucht diese Doktorarbeit wie weit Semidefinite Programmierung in einer Branch-and-Bound Methode zur Lösung des Problems der maximalen stabilen Menge verwendet werden kann.
Kapitel 2 bietet eine Einführung zu den Semidefiniten Programmen an.
Kapitel 3 beschreibt in wenigen Worten wie SDPNAL Semidefinite Programme löst. Lemma 7, der Nachweis vom Lemma 8, die Bemerkung 2 und das Beispiel 2 sind die wichtigsten Beiträge von diesem Kapitel. Das Problem der maximalen stabilen Menge und einige bekannte Tatsachen werden im Kapitel 4 dargestellt. Die Lovász θ-Zahl (4.8) und die Lovász-Schrijver θ'-Zahl (4.9) sind die klassischen Relaxierungen für das Problem der maximalen stabilen Menge. Wir wiederholen einige wohlbekannte alternative Relaxierungen. Im Satz 10, basierend auf M. Laurent [20], wird der Nachweis der Gleichwertigkeit von zwei Formulierungen (4.6) und (4.8) auf den doppelt nichtnegativen Kegel erweitert. Satz 9 stellt einen neuen Nachweis zur Beziehung zwischen (4.6) und einer anderen Relaxierung (4.7) dar. Eine neue Relaxierung von (4.7) wird im Problem (4.18) eingeführt. Nach Proposition 3 sind alle Semidefiniten Relaxierungen in diesem Kapitel mathematisch gleichwertig. Um mögliche Rundungsfehler, die bei die Näherungslösung von SDPNAL auftreten, zu veranschaulichen, ist ein numerischer Vergleich zwischen Relaxierungen (4.7), (4.9) und (4.18) in Tabelle 4.1 angegeben. Die letzte Diskussion in diesem Kapitel wird sich mit der Berechnung einer gültigen oberen Schranke des Optimalwerts von einem primären Semidefiniten Programm aus einer Näherungslösung unter generischen Nichtentartungs-Voraussetzungen befassen.
Die letzten vier Kapitel formulieren und analysieren eine Branch-and-Bound Strategie zur Lösung des Problem der maximalen stabilen Menge. Bei jedem Branching Schritt wird eine Reihe von Semidefiniten Relaxationen gelöst, die zu einer gültigen oberen Schranke für das Problem der maximalen stabilen Menge führen. Wir ergänzen die semidefinite obere Schranke durch Anpassung eines randomisierten Verfahren (von Goemans and Williamson), um lokal optimale Lösungen vom Problem der maximalen stabilen Menge zu generieren. Wir beweisen (Proposition 1), dass die Verteilung zur Erzeugung von Näherungs-lösungen für das Problem der maximalen stabilen Menge nicht von der Faktorisierung der Optimallösung der Semidefiniten Relaxierung abhängt. Die wichtigen Beiträge von diesen Kapiteln werden wie folgt aufgelistet:

1. Um ungenaue Ergebnisse aufgrund von Rundungsfehlern bzw. Fehlern beim Abbrechen des Programms SDPNAL zu beschränken, schlagen wir eine praktische Methode vor, die auf eine Näherungslösung der Lovász-Schrijver θ'-Zahl (4.9) angewendet wird. Die Methode erzeugt eine gültige obere Schranke für das Problem der maximalen stabilen Menge.

2. Wir entwickeln eine neue Methode zur Abschätzung der Größe der maximalen stabilen Menge von einem gegebenen Graph (Abschnitt 6.1). Um diese Methode anzuwenden, nehmen wir an, dass eine obere Schranke für die Größe der maximalen stabilen Menge bekannt ist. Wir führen einen beliebigen $\{0,1\}$-Vektor ein und lösen eine Reihe von Relaxierungen (4.6) für einen Graph definiert durch den $\{0,1\}$-Vektor. Drei verschiedene Methoden werden darstellt, um einen $\{0,1\}$-Vektor zu bilden mit der Absicht die Dimension der Relaxierung (4.6) zu reduzieren (Abschnitt 6.1.1).

3. Eine andere Methode dieser Doktorarbeit ist das Addieren weiterer Nebenbedingungen zur Relaxierung (4.6), um die obere Schranke für das Problem der maximalen stabilen Menge zu stärken und um eine neue Branching Methode zu erhalten. Wir diskutieren wie die Daten der neuen weiteren Nebenbedingungen zu bestimmen sind, damit sich ein tiefer Schnitt ergibt (Abschnitt 6.3). Außerdem gehen wir an, wie die Unzulässigkeit der neuen Relaxierung in Anwesenheit von Rundungsfehlern zu beweisen ist (Abschnitt 6.4).

4. Wir diskutieren den Zusammenhang zwischen dem Verfahren, das in dieser Doktorarbeit vorgeschlagen wurde und dem Verfahren nach Benson und Ye durch Korollar 1 und Bemerkung 14. Lemma 12 (das von Benson und Ye nicht beweisen wurde) bestimmt die Lovász-Zahl der Benson-Ye-Methode. Das letzte Kapitel erlaubt einen Einblick in das praktische Verhalten einer Branch und Bound Methode

Acknowledgement

I would like to express my sincere appreciation to my supervisor Prof. Florian Jarre for his valuable guidance, immeasurable support, encouragement and constructive suggestions which were determinant for the accomplishment of the work present in this thesis. His useful comments and remarks especially during the writing of the thesis is something for that I will always be truly grateful to him.

I also would like to express my gratitude to my co-supervisor Prof. Achim Schädle for his useful and constructive comments, extensive proofreading and editing the entire manuscript.

Also, I extend my gratitude to the rest of my defense committee members, Prof. Stefan Schröer, Prof. Ergon Wanke and Prof. Rüdiger Braun for their direction, dedication and invaluable advice along this project.

At last but not at least, I wish to thank my family for their pure everlasting love and support which encouraged me to keep my hope and strengthened me to fulfill this thesis.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Semidefinite Programming is a subclass of nonlinear optimization problems that can be applied for modeling robust optimization problems, problems in control theory, or subproblems arising from combinatorial optimization problems. Semidefinite programming is an extension of linear programming to the space of real symmetric matrices where the matrices are constrained to be positive semidefinite. The set of positive semidefinite matrices forms a convex cone, so semidefinite programming is a special case of convex conic programming. It contains some well-known optimization problems such as linear and quadratic programming and second order cone programming. A number of interior point methods can be applied to solve semidefinite programs though they can not be solved optimally in polynomial time. Under certain condition, semidefinite programs can be approximated up to a given precision in polynomial time. Software packages with polynomial complexity have been developed to solve broad classes of SDP, *e.g.* SEDUMI [29], SDPT3 [31] or a newer approach SDPNAL [35] for large scale problems, for which a polynomial complexity bound is not known yet.
Semidefinite Programs abbreviated to SDPs have received a great degree of attention because of its many applications to various problems. One of the important applications of SDP is in combinatorial optimization referring to an optimization problem over a discrete structure. So far, search algorithms for combinatorial problems are either not guaranteed to find an optimal solution or not guaranteed to run in polynomial time. Semidefinite programs often provide a strong relaxation of the hard combinatorial problems and hence become an efficient tool for approximating these problems more accurately. A number of NP-hard combinatorial optimization problems can be relaxed to semidefinite programs. Even finding an approximate solution to these problems often is difficult.

Motivated by the recent progress in the solution of large scale semidefinite programs, in particular with the program package SDPNAL, this thesis investigates in how far semidefinite programs can be used in a branch and bound approach for solving the maximum stable set problem. The maximum stable set problem is a classical NP-hard optimization problem which has been studied extensively. The maximum stable set problem has applications in many important practical problems arising, *e.g.* in scheduling, timetabling, molecular biology, coding theory and many other areas [6]. Generally, the maximum stable set problem can be used to solve optimization problems on a given graph where some nodes of the graph conflict each other and a large subset of the vertices is searched such that the subset contains no conflicting vertices. However, there is so far no known exact polynomial time algorithm for this problem.

Following is a short summary of previous research studies on the maximum stable set problem. The Lovász function [21] known as theta number can be considered as one of the famous examples of applying SDP to combinatorial optimization. It provides an upper bound on the stability number of a graph and a lower bound on the chromatic number of the complement graph, known as Lovász sandwich theorem. The *theta number* as the optimum value of a semidefinite program can be computed with an arbitrary precision in polynomial time [24]. An extensive number of research articles are related to the Lovász theta number. Rendl and Dukanovic [7] proposed tightened semidefinite relaxations by adding several types of cutting planes to get more accurate upper and lower bounds. Their observation over a number of numerical examples, however, concluded that the proposed relaxations do not change the value of the Lovász function in a significant way. A hierarchy of semidefinite relaxations of the stability number starting with the theta number was proposed by Monique Lauret [19] and Gvozdenovic et al [13] finding the stability number in a finite number of steps but at a high computational cost. Burer et al. [5] proposed a low rank (specially rank one and two) restriction strategy to the theta number and used continuous optimization techniques to extract large stable sets in fairly large graphs. Various hierarchies of semidefinite relaxations for the maximum stable set problem were proposed by Lovász and Schrijver [22], Sherali and Adams [26] and Lasserre [17]. They proved that these hierarchies can obtain an exact solution to an integer program with n variables taking values in $\{0, 1\}$ after reaching the n^{th} level of the hierarchy. However, their approach results in a sequence of SDP problems of exponentially increasing size. A comprehensive survey to compare these algorithms is provided by Laurent [18]. The MAX-CUT problem is another important combinatorial problem that attracts attention by many researchers over the years and can be relaxed into SDP. Goemans and Williamson [11] proved a polynomial time approximation algorithm for the MAX-CUT problem with strong performance. Offering a performance guarantee makes the algorithm proposed by Goemans and Williamson distinctive. Benson and Ye [3] applied this algorithm to generate an approximation of the solution of a relaxed problem to find a maximum stable set.

Chapter 2

Introduction to Semidefinite Programming

2.1 Preliminaries

Here, we state some standard definitions that are used throughout this paper. The space of real symmetric matrices is denoted by

$$\mathcal{S}^n = \{\ X \mid\ X \in \mathbb{R}^{n\times n},\ X = X^T\},$$

the positive semidefinite cone by

$$\mathcal{S}^n_+ = \{\ X \mid\ X \in \mathcal{S}^n,\ X \succeq 0\}$$

and the cone of nonnegative matrices by

$$\mathcal{N}^n = \{\ X \mid\ X \in \mathcal{S}^n,\ X \geq 0\}$$

where $X \succeq 0$ and $X \geq 0$ indicate that X is a positive semidefinite matrix and a componentwise nonnegative matrix, respectively. $\mathcal{S}^n_+$ is a closed convex cone in $\mathcal{S}^n$ of dimension $n \times (n+1)/2$. The intersection $\mathcal{S}^n_+ \cap \mathcal{N}^n$ is called doubly nonnegative cone.

Definition 1 *The scalar product in SDP denoted by* $\langle \cdot\,,\cdot \rangle : \mathcal{S}^n \times \mathcal{S}^n \to \mathbb{R}$ *is given by*

$$\langle C, X\rangle :=\ C \bullet X :=\ trace(C^T X) := \sum_{i,j} C_{ij}X_{ij}.$$

The definition $trace(C^T X)$ rather than $trace(CX)$ is used since the generalization of this scalar product to rectangular matrices A, B of the same dimension is given by $trace(A^T B)$. The Frobenius norm of a Matrix $X \in \mathbb{R}^{n\times n}$ is defined as $\|X\|_F = \sqrt{\langle X, X\rangle}$.
For given matrices $A^i \in \mathcal{S}^n,\ i = 1, \ldots, m$ we define a linear map $\mathcal{A} : \mathcal{S}^n \to \mathbb{R}^m$ by

$$\mathcal{A}(X) = \begin{bmatrix} A^1 \bullet X \\ \vdots \\ A^m \bullet X \end{bmatrix}. \tag{2.1}$$

The adjoint operator $\mathcal{A}^* : \mathbb{R}^m \to \mathcal{S}^n$ is given by

$$\mathcal{A}^*(y) = \sum_{i=1}^{m} y_i A^i, \; y \in \mathbb{R}^m, \tag{2.2}$$

so that for all X, y, it satisfies $\langle \mathcal{A}^*(y), X\rangle = \langle y, \mathcal{A}(X)\rangle$ by linearity of the trace.

Definition 2 *let $K \subset \mathcal{S}^n$ be a closed convex cone. The dual cone K^D of the cone K is defined by*

$$K^D = \{\; S \in \mathcal{S}^n \;\mid\; \langle S, X\rangle \geq 0 \;\forall X \in K\}.$$

The cone K is called self-dual if $K = K^D$.

It is well-known that the positive semidefinite cone and the cone of nonnegative matrices are self-dual, since by Fejer's theorem for a matrix $X \in \mathcal{S}^n$, $X \succeq 0$ if and only if $\langle X, Y\rangle \geq 0$ for all $Y \succeq 0$. We have the same result for the nonnegative cone. It is easy to show that

$$(\mathcal{S}^n_+ \cap \mathcal{N}^n)^D = \mathcal{S}^n_+ + \mathcal{N}^n. \tag{2.3}$$

A simple generalization of (2.3) is given in the next lemma.

Lemma 1 *Let $I\!\!E$ be a subset of $\{\{i,j\} \mid \; 1 \leq i,j \leq n, \; i \neq j\}$ and $\hat{K}$ be a closed convex cone defined by*

$$\hat{K} = \{\; X \mid \; X \succeq 0, \; X_{ij} \geq 0 \;\; \forall ij \notin I\!\!E\},$$

then the dual cone of $\hat{K}$ is

$$\hat{K}^D = \{\; S_1 + S_2 \mid \; S_1 \succeq 0, \;\; (S_2)_{ij} \geq 0 \;\; \forall ij \notin I\!\!E, \;\; (S_2)_{ij} = 0 \;\; \forall ij \in I\!\!E\}.$$

Proof. In order to keep this derivation self-contained we give a short proof. Let

$$\bar{K} = \{\; S \mid \; S = S_1 + S_2, \; S_1 \succeq 0, \;\; (S_2)_{ij} \geq 0 \;\; \forall ij \notin I\!\!E, \;\; (S_2)_{ij} = 0 \;\; \forall ij \in I\!\!E\}.$$

We show that $\bar{K} = \hat{K}^D$. Let $S \in \bar{K}$. The inclusion $\subseteq$ follows clearly from

$$\langle X, S\rangle \geq 0 \;\text{ for any matrices }\; X \in \hat{K}. \tag{2.4}$$

Conversely, we will show that

$$\langle S, X\rangle \geq 0 \;\; \forall \; X \in \hat{K} \Longrightarrow S \in \bar{K}.$$

Assume $S \notin \bar{K}$, by the separation theorem there exist $\Lambda \in S^n$ such that

$$\inf_{Z \in \bar{K}} \langle \Lambda, Z\rangle > \langle \Lambda, S\rangle.$$

Since $\mathcal{S}^n_+ + \mathcal{N}^n_+$ is a cone, it follows that $\inf_{Z\in\bar{K}} \langle \Lambda, Z\rangle$=0. This implies $\langle \Lambda, S\rangle < 0$ and

$$\langle \Lambda, Z\rangle \geq 0 \;\; \forall \; Z \in \bar{K}.$$

By definition of $\bar{K} \supset \mathcal{S}^n_+$, it follows that $\Lambda \succeq 0$ by Fejer's theorem and also $\Lambda_{ij} \geq 0$ for all $ij \notin I\!\!E$. Hence $\Lambda \in \hat{K}$ which is in contradiction to (2.4). This completes the proof. $\square$

Theorem 1 (***Schur complement***) *Let a matrix A be of the form $\begin{bmatrix} a & \boldsymbol{c}^T \\ \boldsymbol{c} & B \end{bmatrix}$ where $\boldsymbol{c} \in \mathbb{R}^n$, $B \in \mathcal{S}^n$, and a is a real number. Assume that $c \neq 0$. Then the matrix A is positive semidefinite if and only if $a > 0$ and $B - \frac{1}{a}\boldsymbol{c}\boldsymbol{c}^T \succeq 0$. In case that $c = 0$, the matrix A is positive semidefinite if and only if $a \geq 0$ and $B \succeq 0$.*

Proof. The case $c = 0$ is evident. Assume $c \neq 0$, then $a > 0$ must hold. The matrix A can be decomposed as follows

$$\begin{bmatrix} a & \mathbf{c}^T \\ \mathbf{c} & B \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ a^{-1}\mathbf{c} & I \end{bmatrix} \begin{bmatrix} a & 0 \\ 0 & B - a^{-1}\mathbf{c}\mathbf{c}^T \end{bmatrix} \begin{bmatrix} 1 & a^{-1}\mathbf{c}^T \\ 0 & I \end{bmatrix}$$

since the matrix $P := \begin{bmatrix} 1 & 0 \\ a^{-1}\mathbf{c} & I \end{bmatrix}$ is nonsingular, it follows that $A \succeq 0$ if and only if $P^{-1}A(P^{-1})^T \succeq 0$. □

2.2 Convex Conic Programming

Convex conic programming is an important class of optimization problems concerned with the optimization of a linear (convex) objective function over the intersection of an affine space and a convex cone $\mathcal{K}$ by the assumption that both $\mathcal{K}$ and its dual cone $\mathcal{K}^{\mathcal{D}}$ have nonempty interior and are closed. With the definition of Section 2.1, a conic program is defined as

$$\inf_X \{\ C \bullet X \ \mid\ \mathcal{A}(X) = b,\ X \in \mathcal{K}\} \tag{2.5}$$

and its dual is of the form

$$\sup_y \{b^T y \ \mid\ C - \mathcal{A}^*(y) \in \mathcal{K}^D\}. \tag{2.6}$$

For $\mathcal{K} := \mathcal{S}^n_+$, the optimization (2.5) refers to semidefinite programming. It is called linear program if $\mathcal{K} := \mathcal{N}^n$ and doubly nonnegative program if $\mathcal{K} = \mathcal{DNN} := \mathcal{S}^n_+ \cap \mathcal{N}^n$ is the cone of doubly nonnegative matrices. Notice that a linear program (LP) is a special case of a semidefinite program (SDP), however unlike the cone of nonnegative matrices $\mathcal{N}^n$, the cone $\mathcal{S}^n_+$ represented by an infinite number of linear inequalities is non-polyhedral and non-smooth.

2.3 Semidefinite Programming

Semidefinite programming is in fact a special case of cone programming. In this section we repeat some of the properties of the semidefinite programs, the problem (2.5) for $\mathcal{K} := \mathcal{S}^n_+$. The cone $\mathcal{S}^n_+$ represented by an infinite number of linear inequalities is non-polyhedral and non-smooth.

For sake of convenience, we rewrite the standard primal and dual semidefinite program as follows

$$p^* := \inf_X \{ \, C \bullet X \ \mid \ \mathcal{A}(X) = b, \ X \in \mathcal{S}^n_+ \}, \tag{2.7}$$

and

$$d^* := \sup_y \{ b^T y \ \mid \ C - \mathcal{A}^*(y) \in \mathcal{S}^n_+ \}. \tag{2.8}$$

In convex optimization the positive semidefinite constraint arising in (2.8) is named a linear matrix inequality (LMI). A major breakthrough in convex optimization lies in the introduction of interior point methods. These methods were developed in a series of papers and became of true interest in the context of LMI problems following the work of Yurii Nesterov and Arkadii Nemirovsky [24].

Before moving on to the next section we give a brief introduction to the doubly nonnegative program which is obtained if an additional linear conic constraint $X \in \mathcal{N}^n$ (or equivalently $X \geq 0$) is added to SDP. As will be discussed later, the $\mathcal{DNN}$ relaxation provides (theoretically) an improved approximate solution to the maximum stable set problem. $\mathcal{DNN}$ relaxations arise in copositive programming, the optimization (2.5) where $\mathcal{K}$ refers to the copositive cone defined as follows.

Definition 3 *A matrix $M \in \mathcal{S}^n$ is called completely positive if there are k nonnegative vectors $x_1, \ldots, x_k \in \mathbb{R}^n_+$ such that*

$$M = \sum_{i=1}^{k} x_i x_i^T .$$

The cone of completely positive matrices is denoted by $\mathcal{C}^*_n$. It can be shown that the dual cone of $\mathcal{C}^*_n$ called copositive cone $\mathcal{C}_n$ is given by

$$\mathcal{C}_n = \{ M \in \mathcal{S}^n \mid \ x^T M x \geq 0, \ \forall x \geq 0 \}.$$

Clearly, the definition of $\mathcal{C}^*_n$ implies that $\mathcal{C}^*_n \subseteq \mathcal{S}^n_+ \cap \mathcal{N}^n$ and $\mathcal{S}^n_+ + \mathcal{N}^n \subset \mathcal{C}_n$. Copositive programming is NP-hard in general and is not solvable polynomially unless $P = NP$. One way of showing this, is using a theorem by De Klerk [16]. It states that the Lovász-Schrijver relaxation (a $\mathcal{DNN}$ relaxation of the maximum stable set problem described in Chapter 4) returns an exact solution to the maximum stable set problem if the doubly nonnegative cone is replaced by the completely positive cone.

2.4 Properties of SDP

Below we point out some known basic facts in the semidefinite programming context and omit some of the proofs. To see the details one can be referred to [16].

Theorem 2 (***weak duality***) *Let X and y be feasible solutions (satisfying all the linear and nonlinear constraints) of (2.7) and (2.8) respectively. Then the duality gap $\langle C, X \rangle - b^T y$ is nonnegative.*

Proof. There is a well known one line proof. Since $X \succeq 0$ and $C - \mathcal{A}^*(y) \succeq 0$, it follows from Fejer's theorem that

$$0 \leq \langle X, C - \mathcal{A}^*(y)\rangle = \langle X, C\rangle - \langle X, \mathcal{A}^*(y)\rangle = \langle X, C\rangle - \langle y, \mathcal{A}(X)\rangle = \langle C, X\rangle - b^T y.$$

□

Theorem 2 can be generalized to the case of general convex conic programming including doubly nonnegative problems.
Although semidefinite programming is a generalization of linear programming, there are some properties of LP which do not extend to SDP. Unlike linear programming, even if there exist feasible solutions X, y to the primal semidefinite problem (2.7) and and the dual problem (2.8) and if the duality gap is zero, there is no assurance that either (2.7) or (2.8) has an optimal solution. In addition, there is possibly a duality gap at an optimal solution of (2.7) or (2.8). In order to avoid this situation we assume that both primal and dual semidefinite problems satisfy the Slater condition defined below.

Definition 4 *A strictly feasible solution of the* $(semidefinite)$ *program (2.7) is a feasible solution X such that X belongs to the interior of the cone $\mathcal{S}_+^n$. The so-called Slater condition is satisfied if and only if a strictly feasible solution X exists. In this case, the feasible solution X is called Slater point.*

The following theorem provides a guarantee of existence of the primal and dual optimal solution with zero duality gap.

Theorem 3 (***strong duality***) *If a strictly feasible solution of (2.8) (or (2.7)) exists and $d^* < \infty$ (or $p* > -\infty$), the optimal solution of (2.7) (or (2.8)) will be attained and $p^* = d^*$.*

Proof. See [16] □

For completeness we state the conic duality theorem.

Theorem 4 (*Conic duality theorem*) *If there exists an interior feasible solution $X \succ 0$ of (2.7) and a feasible solution of (2.8), then $p^* = d^*$ and the supremum in (2.8) is attained. Similarly, if there exist feasible y, S for (2.8), where $S \succ 0$, and a feasible solution of (2.7), then $p^* = d^*$ and the infimum in (2.7) is attained.*

Note that the scalar product in SDP is the counterpart to the vector inner product in LP. Therefore, we can generalize Farkas' lemma to prove the infeasibility of either primal or dual problems in semidefinite programming. The following lemma based on [16] states:

Lemma 2 *(Primal/Dual infeasibility) Let $\mathcal{A}$ be a linear map and $\mathcal{A}^*$ be the adjoint operator defined by (2.1) and (2.2), respectively. Let b be a real m-dimensional vector. If there exists a vector $\bar{y} \in \mathbb{R}^m$ satisfying the inequalities $\mathcal{A}^*(\bar{y}) \preceq 0$ and $b^T\bar{y} > 0$, the primal problem (2.7) is infeasible. Similarly, the dual problem (2.8) is infeasible if there exists a matrix $\bar{X} \in \mathcal{S}_+^n$ such that $\mathcal{A}(\bar{X}) = 0$ and $C \bullet \bar{X} < 0$.*

Lemma 2 also holds when the positive semidefinite cone $\mathcal{S}_+^n$ is replaced with the doubly non-negative cone $\mathcal{S}_+^n \cap \mathcal{N}^n$.

Lemma 3 *If there is a vector $y \in \mathbb{R}^m$ such that $-\mathcal{A}^*(y) \in \mathcal{S}_+^n + \mathcal{N}^n$ and $b^T y > 0$, then y is an improving ray for the dual problem and the primal problem is infeasible.*

Proof. Let a dual improving ray y be given. If the primal problem has a feasible solution X, it follows that

$$0 < b^T y = y^T \mathcal{A}(X) = \langle \mathcal{A}^*(y), X \rangle = -\langle S_1 + S_2, X \rangle \leq 0,$$

which is a contradiction. □

2.5 Applications of Semidefinite Programming in Combinatorial Problems

A diversity of optimization problems can be reformulated as semidefinite programs. In this section we mention one of the most important applications of SDP in combinatorial optimization.

2.5.1 SDP for the MAX-CUT Problem

One of the applications of semidefinite programs in combinatorial optimization is the Maximum Cut problem. The Maximum Cut problem abbreviated to MAX-CUT problem is known as NP-complete and there is no known polynomial time exact algorithm for these kind of problems. However, it can be relaxed to a semidefinite problem. For graphs with non-negative edge weights Goemans and Williamson [11] proposed a randomized approximation algorithm (summarized in Section 2.5.2 below) which provides a solution with average value of at least 0.87 times the maximum cut value.
Let $G = (V, I\!E)$ be an undirected graph with vertex set V and edge set $I\!E$ that any edge in $I\!E$ is associated with a real nonnegative number w_{ij} known as its weight. The MAX-CUT problem seeks a partition of V into two non-empty disjoint sets V_1 and V_2 whose union is V such that for all edges $(i, j) \in I\!E$ with $i \in V_1$ and $j \in V_2$, the sum of weights w_{ij} is maximized:

$$\max \sum_{\substack{i \in V_1 \\ j \in V_2}} w_{ij}. \tag{2.9}$$

Note that $w_{ij} = w_{ji}$ for any edge $(i, j) \in I\!E$ with $i \in V_1$ and $j \in V_2$, else $w_{ij} = w_{ji} = 0$. We assign a binary vector $x \in \{-1, +1\}^{|V|}$ to the vertex set V with the definition

$$x_i = \begin{cases} 1 & i \in V_1 \\ -1 & i \in V_2. \end{cases}$$

Therefore the constraint $x_i^2 = 1$ for $i \in V$ is the quadratic form of the binary variable $x \in \{-1, +1\}^{|V|}$. By the binary variable x, one can express the MAX-CUT problem (2.9) as:

$$\max \sum_{\substack{i \in V_1 \\ j \in V_2}} w_{ij} = \max_{x \in \{-1,1\}^{|V|}} \frac{1}{2}(\frac{1}{2} \sum_{i,j \in V} w_{ij}(1 - x_i x_j)).$$

Therefore, the Max-Cut problem can be stated as the following quadratic optimization problem:

$$\max \{\frac{1}{4} \sum_{i,j \in V} w_{ij}(1 - x_i x_j) \mid x_i^2 = 1, \ i \in V\} \tag{2.10}$$

which is equivalent to

$$\max \{\frac{1}{4} \sum_{i,j \in V} w_{ij}(1 - X_{ij}) \mid rank(X) = 1, \ X_{ii} = 1, \ i \in V\} \tag{2.11}$$

where $X = xx^T$ with $x \in \{-1, +1\}^{|V|}$. Relaxing the nonlinear rank-1- constraint will lead us to the semidefinite relaxation problem

$$\max \{\frac{1}{4} \sum_{i,j \in V} w_{ij}(1 - X_{ij}) \mid X \succeq 0, \ X_{ii} = 1, \ i \in V\}. \tag{2.12}$$

2.5.2 The Goemans-Williamson Algorithm

Goemans and Williamson proposed a randomized method to generate a $\{-1, 1\}$-vector. Let X be a $n \times n$-symmetric positive semidefinite matrix. Then there is a factorization matrix L such that $X = LL^T$. The randomization method of Goemans and Williamson chooses a random vector r uniformly distributed on the unit sphere of suitable dimension and then assigns

$$\bar{x} = \text{sign}(Lr)$$

and $sign(Lr)$ is the vector whose components are

$$\text{sign}(L_i r) = \begin{cases} 1 & \text{if } L_i r \geq 0 \\ -1 & \text{if } L_i r < 0 \end{cases}$$

where L_i is the i^{th} row of the factor L of X. In this way we obtain random vector $\bar{x} \in \{-1, 1\}^n$. Based on the next two lemmas, they illustrated that applying this algorithm to the optimal solution of the semidefinite relaxation (2.12) generates a cut with an average objective value of at least 87% of maximum cut.

Lemma 4 *Let u and v be unit vectors and r be a uniformly distributed random vector on the unit sphere. Then the expected value of* $\text{sign}(u^T r)\,\text{sign}(v^T r)$ *is given by*

$$E(\text{sign}(u^T r)\,\text{sign}(v^T r)) = \frac{2}{\pi} \arcsin(u^T v).$$

Lemma 5 *Let X^{sdp} be an optimal solution of the semidefinite relaxation (2.12) and let x be a random $\{-1,1\}$-vector generated by the GW-rounding procedure. Then,*

$$E(\frac{1}{4}\sum_{i,j\in V} w_{ij}(1-x_ix_j)) \geq 0,878 \times \frac{1}{4}\sum_{i,j\in V} w_{ij}(1-X^{sdp}_{ij}).$$

Proof. This follows from Lemma 4 and the fact that $\frac{2}{\pi}\arcsin(t) \leq 1+0.878(t-1)$ for $t\in[-1,1]$. □

It is easy to see that the GW-rounding procedure is not dependent on the factorization of X where X is an optimal solution to (2.12). The following proposition states that the factorizations of X are unique up to an orthogonal transformation.

Proposition 1 *Let X be a positive semidefinite matrix. Then there exist matrices B and $\bar{B}$ such that $X = BB^T = \bar{B}\bar{B}^T$. Furthermore, for any such B, $\bar{B}$, there exists an orthogonal matrix Q such that $B = \bar{B}Q$.*

Proof. The existence of factorizations B and $\bar{B}$ is well-known. In order to prove the second part, let $B = U\Sigma V^T$ and $\bar{B} = \bar{U}\bar{\Sigma}\bar{V}^T$ be the singular value decompositions of B and $\bar{B}$. Let the diagonal elements $\sigma_i \geq 0$ and $\bar{\sigma}_i \geq 0$ of the diagonal matrices Σ and $\bar{\Sigma}$ be sorted in decreasing order. It follows $X = BB^T = U\Sigma^2U^T$ and likewise, $X = \bar{U}\Sigma^2\bar{U}^T$. By construction, $\sigma_i^2 = \bar{\sigma}_i^2$ are the eigenvalues of X and the columns u_i of U and $\bar{u}_i$ of $\bar{U}$ are the associated eigenvectors. For sake of notation simplicity, let $\lambda_i := \sigma_i^2$ denote the i^{th} eigenvalue of X.
We recall that for a given eigenvalue of λ_i, the null space of $X-\lambda_iI$ is called the λ_i-eigenspace of X. If the dimension of the λ_i-eigenspace is one, the proof is obvious since the columns of U and $\bar{U}$ are the same up to plus-minus signs. Let the algebraic multiplicity of λ_i be k, $k>1$ and $\lambda_{i-1} > \lambda_i = \lambda_{i+1} = \cdots = \lambda_{i+k-1} > \lambda_{i+k}$. By the definition of eigenspace and $X = U\Sigma^2U^T = \bar{U}\Sigma^2\bar{U}^T$, assume that $\{u^i,\ldots,u^{i+k-1}\}$ and $\{\bar{u}^i,\ldots,\bar{u}^{i+k-1}\}$ are orthonormal bases for the λ_i-eigenspace. Since the eigenspace is uniquely defined, we have $range(u^i,\ldots,u^{i+k-1}) = range(\bar{u}^i,\ldots,\bar{u}^{i+k-1})$. Therefore according to Lemma 6 below, there exists an orthogonal $k\times k$-matrix $\hat{Q}_i$ with

$$\begin{bmatrix} u^i & \ldots & u^{i+k-1}\end{bmatrix} = \begin{bmatrix} \bar{u}^i & \ldots & \bar{u}^{i+k-1}\end{bmatrix}\hat{Q}_i.$$

Since $\lambda_i = \ldots = \lambda_{i+k-1}$, the i^{th} block $U\Sigma$ equals to the i^{th} block of $\bar{U}\Sigma\hat{Q}_i$, *i.e.*

$$\begin{bmatrix} \lambda_i u^i & \ldots & \lambda_{i+k-1}u^{i+k-1}\end{bmatrix} = \begin{bmatrix} \lambda_i\bar{u}^i & \ldots & \lambda_{i+k-1}\bar{u}^{i+k-1}\end{bmatrix}\hat{Q}_i. \tag{2.13}$$

Let X have l distinct eigenvalues, then (2.13) implies $U\Sigma = \bar{U}\Sigma\hat{Q}$ where $\hat{Q}$ is a l-block diagonal matrix. This completes the proof. □

Let Q be an orthogonal matrix. Evidently, if r is uniformly distributed with $\|r\|_2 = 1$, the same is true for Qr. Therefore according to Proposition 1, the GW-procedure uses the same

distribution for B and $\bar{B}$ but different random events.
The following lemma is presented in [16] without proof. The proof of this lemma is quite straightforward, but we give it for completeness.

Lemma 6 *Let U and $\bar{U}$ be $m \times n$-orthogonal matrices where $m > n$, such that $C(U) = C(\bar{U})$ where $C(\cdot)$ denotes the column space. Then, there exists an orthogonal matrix Q such that $UQ = \bar{U}$.*

Proof. Let $S = [u^1, \ldots, u^n, v^{n+1}, \ldots, v^m]$ form a basis for $\mathbb{R}^m$ where u^i is the i^{th} column of U. An orthogonal matrix $\hat{U} := \begin{bmatrix} U & \tilde{U} \end{bmatrix}$ such that $\tilde{U} \perp C(U)$ can be constructed by applying the Gram-Schmidt procedure to the matrix S. Since $C(U) = C(\bar{U})$, in the same way the orthogonal matrix $\hat{V} := \begin{bmatrix} \bar{U} & \tilde{U} \end{bmatrix}$ can be formed. Set $\hat{Q} := \hat{U}^T \hat{V}$. By the definition of $\hat{Q}$ we have

$$\hat{Q} = \begin{bmatrix} \hat{Q}_{11} & \hat{Q}_{12} \\ \hat{Q}_{21} & \hat{Q}_{22} \end{bmatrix} = \begin{bmatrix} U^T \bar{U} & U^T \tilde{U} \\ \tilde{U}^T \bar{U} & \tilde{U}^T \tilde{U} \end{bmatrix}.$$

The orthogonality of $\hat{U}$ and $\hat{V}$ implies that $\hat{Q}_{12} = \hat{Q}_{21} = 0$ and $\tilde{U}^T \tilde{U} = I$. Then, we have

$$\hat{Q} = \begin{bmatrix} U^T \bar{U} & 0 \\ 0 & I \end{bmatrix}.$$

Since $\hat{Q}$ is orthogonal, $\hat{Q}$ turns out to be a block diagonal orthogonal matrix. Therefore, there exists an orthogonal matrix Q with $Q = U^T \bar{U}$ which ends the proof. $\square$

Chapter 3

The Complexity of Solving Semidefinite Programs

The practical aspect of semidefinite programming is that SDP can be solved in a certain sense in polynomial time as follows from the results given by Nestrov and Nemirovsky [24]. The solvability of SDP in polynomial time is under the assumption that the solution set of the SDP is contained in a ball, the radius of which can be represented by polynomially many digits. Example 1 below provides a counter example of solvability of SDPs with polynomial complexity in general. But first we recall the definition of polynomial algorithm.
Algorithms can be classified by the amount of time they need to complete compared to their input size. Following Grötschel et al. [12] an algorithm is said to run in polynomial time, or equivalently a problem is solved in polynomial time, if there exist a polynomial $p()$ such that the time taken by the algorithm to solve the problem is bounded above by $p(size(input))$.
To calculate the time taken to solve, we normally evaluate the number of steps the procedure takes to solve the problem, assuming each step takes unit time. In the arithmetic model of computation all basic operations (addition, multiplication, comparison) take a unit time step to perform, regardless of the sizes of the operands.
When we are to measure the size of the input, we need to take two factors into account. The first is the number of objects that must be provided for the input. For instance, for a sorting program, the size of the input is the number of elements to be sorted. The second is the size of each input value known as coding length. It also contributes to the amount of the input data. For this reason, in order to measure the problem size or the input length, one fixes an encoding scheme. An algorithm is polynomial time if the running time is bounded by $p(n)$ where n denotes the input length [10].

Example 1 *Consider the optimization problem*

$$\min \{ \; x_n \mid \; x_0 = 2, \;\; x_{i-1}^2 - x_i \leq 0, \;\; i = 1, \ldots, n \;\; x_i \in \mathbb{R}\}.$$

This problem can be transformed to a semidefinite program if we replace the inequality constraint by the semidefinite constraint

$$\begin{bmatrix} 1 & x_{i-1} \\ x_{i-1} & x_i \end{bmatrix} \succeq 0, \quad i = 1, \dots, n. \tag{3.1}$$

Starting with $x_0 = 2$, it is obvious that the variable x_n attains its minimum value at at least 2^{2^n}. It is possible to compute 2^{2^n} with n multiplications. However the space needed to encode 2^{2^n} is 2^n digits, thus the space used to represent it is exponential rather than polynomial. An algorithm of exponential complexity requires an exponentially increasing amount of time and computer memory for only a constant increase in problem size. Therefore, this kind of algorithm typically become impractical to carry out.

A number of algorithms based on interior point methods (IPMs) have been proposed to solve semidefinite programs. IPMs are of interest because they have application to several classes of convex optimization, such as linear and semidefinite programming as two well known subclasses of conic optimization. In addition, they have polynomial complexity for computing an ϵ-approximation to an optimal solution. Todd and Nemirovsky [23] provide a comprehensive survey on interior point methods. Although IPMs have proven reliable and efficient on small and medium sized semidefinite programs, the computational and storage needs of these methods can limit the size of problems that can be solved. The interior point linear system that arises when solving (2.7) can be reduced to a linear system in m equations and m unknowns. Unfortunately, even when $\mathcal{A}$ is sparse, the $m \times m$ systems typically are not so. This limits the application of IPMs to instances with small m, say $m \leq 5000$.

3.1 Augmented Lagrangian Function

The method of augmented Lagrangian function provides a way to deal with constrained optimization problems using algorithms for unconstrained problems. Consider the nonlinear optimization problem

$$\min \{ \ f(x) \mid \ f_i(x) \leq 0, \quad 1 \leq i \leq m \ \} \tag{3.2}$$

where f, $f_i : \mathbb{R}^n \to \mathbb{R}$ belong to the class of C^2 (the space of twice continuously differentiable functions). Let $r > 0$. The standard augmented Lagrangian function for (3.2) is defined as follows:

$$\mathcal{L}(x, y, r) := f(x) + \frac{r}{2} \sum_{i=1}^{m} ((f_i(x) + \frac{y_i}{r})_+)^2 - \frac{1}{2r} \sum_{i=1}^{m} y_i^2, \tag{3.3}$$

where the projection map onto the positive orthant is denoted by $(.)_+$. Let $\tilde{x}$ be a minimizer of $\mathcal{L}(., y, r)$ and $f_i = f_i(\tilde{x})$. Then, the expression $\tilde{y}_i := (f_i + \frac{y_i}{r})_+$ may be considered as an update of the Lagrange multiplier associated with $\tilde{x}$ and r.

The augmented Lagrangian function is a combination of Lagrangian function with penalty terms. The objective function $f(x)$ is penalized with squared penalty terms to keep the solution near the feasible set. Let $\bar{x}$ be an optimal solution of (3.2) and $\bar{y}$ be an associated multiplier. Thus the complementary slackness $\bar{y}_i f(\bar{x}_i) = 0$ is satisfied and obviously $\mathcal{L}(\bar{x}, \bar{y}, r) = f(\bar{x})$.

Algorithm1: given $y^0 \geq 0$, $r > 0$

```
1. $x^{k+1} := \text{argmin}_x \ \mathcal{L}(x, y^k, r)$;
2. Update $y$:
   $y_i^{k+1} := (f_i(x^{k+1}) + \frac{y_i^k}{r})_+$  for $i = 1 \ldots, m$;
3. $r = r\sigma_k$  for some factor $\sigma_k \geq 1$.
```

Here, y_i^{k+1} denotes the i^{th} component of y at step $k+1$. According to the following theorem for sufficiently large r, the augmented Lagrangian method locally converges to the optimal solution of (3.2) under the strict complementary condition $\bar{y}_i - f(\bar{x}_i) > 0$ and second order sufficient condition [15]. The existence of a local optimal solution in Step 1 follows from the local convexity of $\mathcal{L}$ with respect to x.

Theorem 5 *Let $(\bar{x}, \bar{y})$ be a KKT-point of (3.2) satisfying the strict complementary condition and second order sufficient condition. There is a neighborhood of $\bar{y}$ with radius δ and of $\bar{x}$ with radius ϵ, denoted by $B_\delta(\bar{y})$ and $B_\epsilon(\bar{x})$, such that for all $y \in B_\delta(\bar{y})$ there is a $x(y) \in B_\epsilon(\bar{x})$ such that $x(y)$ is a strict local minimum of $\mathcal{L}(x, y, r)$ for large enough r. Moreover, $\mathcal{L}(x(y), y, r)$ is concave for all $y \in B_\delta(\bar{y})$ where $\bar{y}$ is the strict local maximum of $\mathcal{L}(x(y), y, r)$.*

3.2 Semismooth Newton Augmented Lagrangian

The algorithm of the augmented Lagrangian function can be generalized towards solving SDPs. The SDPNAL solver introduced by Xin-Yuan Zhao et al. [35], is an up-to-date Matlab software for semidefinite programming based on the semi-smooth Newton-conjugate gradient (CG) augmented Lagrangian method. They represent the positive semidefinite constraint implicitly by using a projection operator and a semismooth Newton approach combined with the conjugate gradient method is proposed to minimize the dual augmented Lagrangian function. Below we provide a brief description of how SDPNAL works, for details the reader may be referred to [35]. This algorithm is intended for solving the following SDP problem:

$$\min \ \{b^T y \mid \mathcal{A}^* y - C \succeq 0\} \tag{3.4}$$

Here, we follow the notation of [35] where the dual (2.8) is written as a minimization problem. Apart from sign changes, the duality theory of Section 2.4 also applies to (3.4). The augmented Lagrangian function for (2.8) is defined as

$$\mathcal{L}(X, y, \sigma) := b^T y + \frac{1}{2\sigma}(\|\Pi_{\mathcal{S}_+^n}(X - \sigma(\mathcal{A}^*(y) - C))\|^2 - \|X\|^2), \tag{3.5}$$

where $\sigma > 0$ is a penalty parameter and $\Pi_{\mathcal{S}^n_+}(X)$ denotes the orthogonal projection of X onto $\mathcal{S}^n_+$. The variable X in (3.5) corresponds to y in (3.3) while y corresponds to x in (3.3). SDPNAL generates two sequences $\{y^k\}$ and $\{X^k\}$ converging to the optimal solutions of (3.4) and its dual problem under the Slater condition for both problems ([35], Theorem 4.1).

Algorithm2:
Let an initial solution $X^0 \in \mathcal{S}^n$, a penalty parameter σ_0 and $\rho > 1$ be given,

```
1. $y^{k+1} \leftarrow \text{argmin } \mathcal{L}(X^k, y, \sigma_k)$.
2. $X^{k+1} \leftarrow \Pi_{\mathcal{S}^n_+}(X^k - \sigma_k(\mathcal{A}^*(y^{k+1}) - C))$.
3. $\sigma_{k+1} = \rho\sigma_k$ or $\sigma_{k+1} = \sigma_k$.
```

Above, σ_k can be interpreted as a penalty parameter, and in Step 3 this parameter is either increased or it is kept of its old value. The first step of Algorithm 2 is to find a solution y minimizing the unconstrained convex function (3.5). To simplify the notation, set $\phi(y) := \mathcal{L}(y, X, \sigma)$. The first order condition implies

$$\nabla\phi(y) = 0, \tag{3.6}$$

where $\nabla\phi(y) = b - \mathcal{A}\Pi_{\mathcal{S}^n_+}(X - \sigma(\mathcal{A}^*(y) - C))$. Since $\Pi_{\mathcal{S}^n_+}(.)$ is Lipschitz continuous, according to Rademacher's theorem (Theorem 6 below) the function $\nabla\phi$ is almost everywhere differentiable. Referring to Theorem 3.4 in [35], there exists a minimal solution y to (3.6).
Since $\Pi_{\mathcal{S}^n_+}(.)$ is strongly semismooth [30], a Newton-like optimization algorithm called semismooth Newton method can be applied to the nonsmooth equation (3.6). In order to solve (3.6), the semismooth Newton method can be defined generally as follows: given a vector y^k, compute y^{k+1} by

$$y^{k+1} = y^k + \alpha_k d^k$$

where $d^k = -V_k^{-1}\nabla\phi(y^k)$, $V_k \in \partial(\nabla\phi(y))$ where $\partial(\nabla\phi(y))$ denotes the Clarke's generalized Jacobian of $\nabla\phi$ (see Definition 5 below) and the step size is controlled by the choice of α_k. For further details, the reader may be referred to [35].
If the dual problem of $\min_y \ \mathcal{L}(y, X, \sigma)$ satisfies the Slater condition, the sequence y^k generated by the proposed semismooth Newton method converges to an optimal solution of the problem (3.6) [[35], Theorem 3.4].
The second step is to update the Lagrange multiplier X^{k+1}. Since y^{k+1} is the global minimizer of $\mathcal{L}(X^k, y, \sigma_k)$, then $\nabla\mathcal{L}_y(X^k, y^{k+1}, \sigma_k) = 0$ leads us to a solution y^{k+1} that is feasible for the dual problem of (3.4).
Finally, the updated penalty parameter σ_{k+1} is obtained in the third step.
In what follows, we introduce some concepts related to the SDPNAL method.

3.2.1 Some Facts of Lipschitz Continuous Functions

The following well-known theorem states the differentiability of a Lipschitz function [8].

Theorem 6 *(Rademacher's theorem) If a function $f : \mathbb{R}^n \to \mathbb{R}^m$ is Lipschitz continuous, then f is differentiable almost everywhere.*

Rademacher's theorem can be directly extended from vector functions to matrix functions. Definition 5 and Proposition 2 are quoted from [32].

Definition 5 *Let $V \subset \mathbb{R}^n$ be an open set and the function $f : V \to \mathbb{R}^m$ be Lipschitz continuous in some neighborhood of $x \in V$. Let D_f denote the set of points at which f admits a derivative $\nabla f(x) \in \mathbb{R}^{m\times n}$. Clarke's generalized Jacobian is defined as*

$$\partial f(x) = co(\{G \in \mathbb{R}^{m\times n} : \exists x_k \subset D_f \quad with\ x_k \to x, \nabla f(x_k) \to G\})$$

where co denotes convex hull.

Proposition 2 *Let $f : V \to \mathbb{R}^m$ be defined on the open set $V \subset \mathbb{R}^n$. Then for $x \in V$, the function f is semismooth at x if and only if f is Lipschitz continuous near x, directionally differentiable at x and*

$$\sup_{G\in\partial f(x+s)} \|f(x+s) - f(x) - Gs\| = o(\|s\|) \quad as \quad s \to 0.$$

Let $\phi : \mathbb{R}^n \to \mathbb{R}$ be a continuously differentiable function. The characteristic equation below is known for a differentiable function:

$$\phi(x + t\Delta x) = \phi(x) + D\phi(x)[t\Delta x] + o(t),$$

where $D\phi(x)[\ .\ ] : \mathbb{R}^n \to \mathbb{R}$ is linear and

$$D\phi(x)[\Delta x] \equiv \nabla\phi(x)^T \Delta x \equiv \langle\nabla\phi(x), \Delta x\rangle \in \mathbb{R}.$$

The linearity of $D\phi(x)[\ .\ ]$ identifies $\nabla\phi(x)$ as the element of the Euclidean space $\mathbb{R}^n$ and therefore the map $x \to \nabla\phi(x)$ is of the form $\mathbb{R}^n \to \mathbb{R}^n$. Likewise the same discussion leads us to the fact that the map $x \to \nabla^2\phi(x)$ is of the form $\mathbb{R}^n \to \mathbb{R}^{n\times n}$. There are two interpretations of $\nabla^2\phi(x)$, but first we recall the definition of a bilinear form.

Definition 6 *Let $\mathbb{E}$ be an Euclidean space. The map $\Phi : \mathbb{E} \times \mathbb{E} \to \mathbb{R}$ is called bilinear if*

$$\Phi(\alpha v_1 + \beta v_2, u) = \alpha\Phi(v_1, u) + \beta\Phi(v_2, u)$$

and

$$\Phi(v, \alpha u_1 + \beta u_2) = \alpha\Phi(v, u_1) + \beta\Phi(v, u_2).$$

Equivalently, $\Phi(v, u)$ is linear in each argument v and u.

To continue the previous discussion, $\nabla^2\phi(x)$ can be described in two ways:
i) $\nabla^2\phi(x) : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ is a bilinear form. If y,$z \in \mathbb{R}^n$, then

$$\nabla^2\phi(x)[y,z] = y^T\nabla^2\phi(x)z$$

ii) $\nabla^2\phi(x) : \mathbb{R}^n \to \mathbb{R}^n$ is a linear mapping in the characteristic equation

$$\nabla\phi(x+ty) = \nabla\phi(x) + \nabla^2\phi(x)[ty] + o(t)$$

for $y \in \mathbb{R}^n$.

Lemma 7 *If a function $f : \mathbb{R}^n \to \mathbb{R}^n$ is Lipschitz continuous with modulus $L \leq 1$ and differentiable at a point x, then*

$$|\lambda| \leq 1 \tag{3.7}$$

where λ is any eigenvalue of $Df(x)$.

This theorem is well-known. For completeness, a short proof is given.

Proof. Since f is Lipschitz, Rademacher's theorem implies that f is differentiable almost everywhere. Let $x \in \mathbb{R}^n$ be a point where $Df(x)$ exist. The characteristic equation

$$f(x+t\Delta x) = f(x) + tDf(x)[\Delta x] + o(t)$$

is known for the differentiable function f. The derivative $Df(x)$ is represented in coordinates by the Jacobian matrix. By contradiction we assume that there exist an eigenvalue λ such that $|\lambda| > 1$. Let Δx be the eigenvector corresponding to λ. By definition of Lipschitz function we have

$$\|\frac{f(x+t\Delta x) - f(x)}{t}\| \leq L\|\Delta x\|.$$

On the other hand

$$\|\frac{f(x+t\Delta x) - f(x)}{t}\| = \|\frac{tDf(x)[\Delta x] + o(t)}{t}\| = \|\lambda\Delta x + \frac{o(t)}{t}\|.$$

Then

$$\lim_{t\to 0} \|\lambda\Delta x + \frac{o(t)}{t}\| = |\lambda|\|\Delta x\|$$

contradicts the assumption. □

The following theorem proposed by J. Moreau (Theorem 3.2.5 in [14]) provides a generalized form of decomposition of an arbitrary vector into two orthogonal vectors.

Theorem 7 *Let K be a closed convex cone and K^* be the polar cone of K, i.e. $K^* = \{z \in \mathbb{R}^n \mid \langle z, x\rangle \leq 0 \ \forall x \in K\}$. For an arbitrary $x \in \mathbb{R}^n$, the following statements are equivalent:*
i)) $x = x_1 + x_2$ with $x_1 \in K$ and $x_2 \in K^*$ such that $\langle x_1, x_2\rangle = 0$.
ii)) $x_1 = \Pi_K(x)$ and $x_2 = \Pi_{K^*}(x)$.

The following well-known theorem says that the pointwise supremum of many convex functions is convex, see *e.g.* [15].

Theorem 8 *If $f_i : \mathbb{R}^n \to \mathbb{R}$ are convex for $i \in I$ (the set I could be infinite). Then*

$$f(x) = \sup_{i \in I} f_i(x)$$

is convex on its domain $D_f = \cap_{i \in I} D_{f_i} \cap \{x \mid \sup_i f_i(x) < \infty\}$.

The convergence of SDPNAL can be discussed based on the convexity and concavity property of the augmented Lagrangian function (3.5).

Lemma 8 *The augmented Lagrangian function $\mathcal{L}(X, y, \sigma)$ is convex with respect to y and concave with respect to X.*

Remark 1 *The convexity of $\mathcal{L}(X, y, \sigma)$ with respect to y is derived in [35] based on the following definition*

$$\mathcal{L}(X, y, \sigma) = \max_{Z \in \mathcal{S}^n_+} \{b^T y - \langle Z, \mathcal{A}^* y - C\rangle - \frac{1}{2\sigma}\|Z - X\|^2\} \tag{3.8}$$

from [25] and Theorem 8 above. Below, we establish the concavity of $\mathcal{L}(X, y, \sigma)$ with respect to X. For completeness, we also derive the equality (3.8).

Proof. In order to establish the equality (3.8), let us use the abbreviation $M := X - \sigma(\mathcal{A}^* y - C)$ and set $\Phi(X) := \max_{Z \in \mathcal{S}^n_+} \{b^T y - \langle Z, \mathcal{A}^* y - C\rangle - \frac{1}{2\sigma}\|Z - X\|^2\}$.

$$\begin{aligned}
\Phi(X) &= b^T y + \max_{Z \in \mathcal{S}^n_+} \{-\langle Z, \mathcal{A}^* y - C\rangle - \frac{1}{2\sigma}\|Z - X\|^2\} \\
&= b^T y + \frac{1}{2\sigma} \max_{Z \in \mathcal{S}^n_+} \{\langle -2\sigma Z, \mathcal{A}^* y - C\rangle + 2\langle Z, X\rangle - \|Z\|^2 - \|X\|^2\} \\
&= b^T y - \frac{1}{2\sigma}\|X\|^2 + \frac{1}{2\sigma} \max_{Z \in \mathcal{S}^n_+} \{-\|Z\|^2 + 2\langle M, Z\rangle - \|M\|^2 + \|M\|^2\}) \\
&= b^T y - \frac{1}{2\sigma}\|X\|^2 + \frac{1}{2\sigma}(\|M\|^2 - \min_{Z \in \mathcal{S}^n_+} \{\|Z - M\|^2\}) \\
&= b^T y - \frac{1}{2\sigma}\|X\|^2 + \frac{1}{2\sigma}(\|M\|^2 - \|\Pi_{\mathcal{S}^n_+}(M) - M\|^2).
\end{aligned}$$

Decomposition of $M := M_1 + M_2$ based on Theorem 7 implies

$$\|M\|^2 - \|\Pi_{\mathcal{S}_n^+}(M) - M\|^2 = \|\Pi_{\mathcal{S}^n_+}(M)\|^2.$$

This proves the equality (3.8). The concavity of $\mathcal{L}(X, y, \sigma)$ with respect to X is shown as follows: Take the derivative of $\mathcal{L}$ defined in (3.5) with respect to X,

$$\nabla_X \mathcal{L}(X, y, \sigma) = \frac{1}{\sigma}(\Pi_{\mathcal{S}^n_+}(X - \sigma(\mathcal{A}^* y - C)) - X).$$

Since $\Pi_{\mathcal{S}^n_+}$ is Lipschitz and a semismooth function, Lemma 7 implies that the absolute value of all eigenvalues of $\nabla_X \Pi_{\mathcal{S}^n_+}(X - \sigma(\mathcal{A}^* y - C))$ is at most 1 wherever they are defined.. Therefore the eigenvalues of $\nabla^2_X \mathcal{L}(X, y, \sigma)$ are nonpositive and $\nabla^2_X \mathcal{L}(X, y, \sigma) \preceq 0$ wherever $\nabla^2_X \mathcal{L}$ is defined.

□

Remark 2 *By definition of $\mathcal{L}$ in (3.5) and the reformulation below*

$$\frac{1}{2\sigma}\|\Pi_{\mathcal{S}^n_+}(X - \sigma(\mathcal{A}^*(y) - C))\|_F^2 = \frac{1}{2\sqrt{\sigma}}\|\Pi_{\mathcal{S}^n_+}(C + \frac{1}{\sigma}X - \mathcal{A}^*(y))\|_F^2,$$

it follows that the convexity of $\mathcal{L}$ with respect to y is equivalent to the convexity of the function $y \longmapsto \|\Pi_{\mathcal{S}^n_+}(-\mathcal{A}^*(y) + \tilde{C})\|$ for the matrix $\tilde{C} = C + \frac{1}{\sigma}X$. The next example shows that the function $\phi(x) = \|\Pi_K(x)\|$ is not convex in general even if K is a convex set.

Example 2 *Let $K = \{x \in \mathbb{R}^2 \mid \|x - \begin{bmatrix} 0 \\ 1 \end{bmatrix}\| \leq 1\}$ and $y := \begin{bmatrix} -m \\ 2 \end{bmatrix}$ and $z := \begin{bmatrix} m \\ 2 \end{bmatrix}$. For $m \to +\infty$, $\Pi_K(y) \to \begin{bmatrix} -1 \\ 1 \end{bmatrix}$ and $\Pi_K(z) \to \begin{bmatrix} 1 \\ 1 \end{bmatrix}$. Assume $\lambda = \frac{1}{2}$, then $2 = \|\Pi_K(\frac{y}{2} + \frac{z}{2})\|$ but $\frac{1}{2}\|\Pi_K(y)\| + \frac{1}{2}\|\Pi_K(z)\| \to \sqrt{2}$ when $m \to +\infty$.*

A similar example can be constructed in case of having a convex cone K to show that the convexity does not hold for $\|\Pi_K(x)\|$.

Lemma 9 *Let K be a closed convex set in $\mathbb{R}^n$. For $y \in \mathbb{R}^n$, the map $\phi : \mathbb{R}^n \to \mathbb{R}$ defined as $\phi(y) := \|y - \Pi_K(y)\|$ is convex.*

Proof. It is easy to show the convexity by the direct definition. Let $\bar{x} := \Pi_K(x)$ and $\bar{y} = \Pi_K(y)$.

$$\begin{aligned} \|(\lambda x + (1-\lambda)y) - \Pi_K(\lambda x + (1-\lambda)y)\| &\leq \|(\lambda x + (1-\lambda)y) - (\lambda \bar{x} + (1-\lambda)\bar{y})\| \\ &\leq \lambda\|x - \bar{x}\| + (1-\lambda)\|y - \bar{y}\|. \end{aligned}$$

□

Chapter 4

Maximum Stable Set

Here we describe some terminologies. An undirected graph G is a pair of sets $(V, \mathbb{E})$ where V is the vertex set of G and $\mathbb{E} \subset \{\{i, j\} \mid i, j \in V\}$ is the edge set of G. For an edge $\{i, j\} \in \mathbb{E}$, the vertices i and j are called end vertices. The edge with equal end vertices is a loop and the edges that have the same end vertices are parallel. A simple graph is a graph with no loops and no parallel edge. The degree of a vertex of a graph is the number of edges connected to the vertex. The complement of a simple graph $G = (V, \mathbb{E})$ is the simple graph $\bar{G} = (V, \bar{\mathbb{E}})$, where the edges in $\bar{\mathbb{E}}$ are exactly the edges not in G. A subgraph of G is a graph $G' = (V', E')$ where $V' \subseteq V$ and $E' \subseteq \{\{i, j\} \mid i, j \in V'\}$. G' is said to be the subgraph induced by V' if $E' \subseteq E$ includes exactly those edges $\{i, j\} \in E$ for which i,$j \in V'$. In this dissertation, all subgraphs are induced unless otherwise noted. The adjacency matrix of G is denoted by A_G of dimension $|V| \times |V|$. Its $(i, j)^{th}$ entry is 1 if the i^{th} vertex and the j^{th} vertex are adjacent, *i.e.* if there is an edge connecting them. A graph G is complete if all its vertices are pairwise adjacent. A complete graph with n nodes is denoted by $\mathcal{K}_n$. A bipartite graph is a simple graph in which the node set V can be partitioned into two disjoint sets V_1 and V_2 such that any node $v \in V_1$ (*or* V_2) may be adjacent only to vertices in V_2 (*or* V_1).
A clique C is a subset of V such that the graph defined by the pair $(C, \mathbb{E} \cap \{\{i, j\} \mid i, j \in C\})$ is complete. The clique number $\omega(G)$ is the cardinality of a largest clique in G. A stable set in a graph is a set of vertices $S \subseteq V$ with the property that the vertices of S are pairwise non adjacent. Independent set is an equivalent word for stable set found in some contexts. The *stability number* $\alpha(G)$ is the cardinality of a largest stable set in G. A vertex cover of an undirected graph G is a subset V' of V such that every edge in $\mathbb{E}$ is adjacent to at least one node $v' \in V'$. It is well known that the Maximum Stable Set problem on G is equivalent to the minimum vertex cover problem on G and to the maximum clique problem on the complement graph of G, *i.e.* S is an independent set of G if and only if S is a clique of the complement graph of G and if and only if $V \setminus S$ is a vertex cover of G, see *e.g.* [34]. Any results obtained for one of these problems has its equivalent forms for the other problems. Furthermore, these problems are NP-complete problems on arbitrary graphs. A common way to cope with NP-hardness of

a problem is to devise algorithms that give approximate solutions. The *chromatic number* of a given graph G is the minimal number of colors needed to color the nodes such that no two adjacent nodes share the same color. The graph with chromatic number k is k-colorable. The chromatic number is at least the quotient $\frac{|V|}{\alpha(G)}$. To prove this, let the vertex v be assigned color 1. Since any two adjacent vertices do not color the same, all nonadjacent nodes to v form a stable set that can be colored by color 1. In other words, the nodes of each stable set α_i have the same color. A partiton of V into stable sets $\alpha_1, \ldots, \alpha_k$ implies that G is k-colorable. On the other hand we have $\sum \alpha_i = |V| \leq k\alpha(G)$.
The all ones vector is denoted by e, and the all-ones-matrix by $J = ee^T$ and the identity by I. We denote by e_i the i^{th} column of the identity matrix I of appropriate dimension. For a matrix $A \in \mathcal{S}^n$, $diag(A)$ denotes a vector in $\mathbb{R}^n$ whose entries are the diagonal entries of A.

Definition 7 *Let $G = (V, I\!\!E)$ be a simple and undirected graph. A locally maximal stable set of G is a set $S \subset V$ with the property that adding any vertex to S will destroy the stable set property.*

A simple greedy heuristic in order to find a locally maximal stable set is given below:

```
GREEDY
Input: given a graph $G = (V, I\!\!E)$ and $S = \emptyset$
$i$) Find the node $v$ of minimum degree in $G$, $S \leftarrow S \cup \{v\}$
$ii$) Delete $v$ and its neighbors from $G$
$iii$) If $V = \emptyset$, end; else go to ($i$).
```

4.1 Stable Set Polytope

In this section we review some relaxations of the maximum stable set and the facts relating to it. Let $G = (V, I\!\!E)$ be a simple and undirected graph. Let $\mathcal{X}^S \in \mathbb{R}^{|V|}$ be the characteristic vector (or incidence vector) of a stable set S with the following components:

$$\mathcal{X}_i^S = \begin{cases} 1 & i \in S \\ 0 & else. \end{cases}$$

The convex hull of the incidence vectors of all stable sets S of G is called stable set polytope and denoted by

$$STAB(G) := conv(\mathcal{X}^S \mid \; S \subseteq G \;\; stable \;\; set). \tag{4.1}$$

Let us remind that the set of convex combinations of some points is called a convex hull and is written as $conv(x_1, \ldots, x_n)$. A polyhedron is the intersection of a finite number of half spaces. If a polyhedron is bounded, we call it polytope. Since the solution of a linear program over a bounded domain is always attained at a vertex, the following linear program determines the maximum stable set,

$$\alpha(G) := \max \{e^T x \mid \; x \in STAB(G)\}. \tag{4.2}$$

The stable set problem can be reformulated as an integer linear program by adding the so called edge-inequality constraints

$$x_i + x_j \leq 1 \ \ for \ \ all \ \ (i,j) \in I\!\!E \tag{4.3}$$

where the vector x has a binary domain as follows:

$$\alpha(G) = \max \ \{e^T x \mid \ x_i \in \{0,1\}, \ x_i + x_j \leq 1 \ \forall (i,j) \in I\!\!E\}. \tag{4.4}$$

As with many other problems of combinatorial optimization, using the appropriate formulation of the maximum stable set problem is of crucial importance in solving the problem. A linear relaxation of (4.4) can be obtained if the binary constraint is replaced by the continuous constraint $0 \leq x_i \leq 1$. The nonnegativity constraint and the edge inequality (4.3) construct a linear relaxation of $STAB(G)$, denoted by $FRAC(G)$, and another relaxation is obtained by the nonnegativity constraint and the clique inequality

$$\sum_{i \in Q} x_i \leq 1 \ \ \forall \ Q \ clique \ in \ G,$$

denoted by $QSTAB(G)$. By this construction, we get

$$STAB(G) \subseteq QSTAB(G) \subseteq FRAC(G).$$

It is easy to see that a vector x with a positive integer domain (or more precisely, binary domain) satisfying the edge inequality induces a stable set and *vice versa*. The relaxation $FRAC(G)$ provides an exact solution to the maximum stable set problem if and only if G is a connected bipartite graph [1]. In this case the maximum stable set problem can be solved in polynomial time. In general, the linear relaxation of (4.4) is weak and not appropriate to approximate the stable set polytope (4.1). To see this, consider the complete graph $\mathcal{K}_n$. The vector $(\frac{1}{2}, \ldots, \frac{1}{2})^T$ is the optimal solution of the relaxation (4.4). However, this vector is not in $STAB(G)$ since it can not be obtained as a convex combination of incidence vectors of the stable sets of $\mathcal{K}_n$. The only multiple of e contained in $STAB(G)$ is $\frac{1}{n}e$. Besides the linear relaxation of (4.4) for the maximum stable set problem, we can also find other relaxations in the literature [12]. We point out that all well-known linear relaxations of (4.2) are based on necessary conditions for the stable set property.

4.2 Semidefinite Relaxation of the Maximum Stable Set Problem

A well-known equivalent formulation of the *stability number* $\alpha(G)$ can be characterized by the following integer nonlinear program which will be used to infer a semidefinite relaxation of the stable set problem:

$$\begin{aligned}\alpha(G) &= \max \{\ e^T x \ \mid\ x \in \{0,1\}^{|V|},\ \ x_i x_j = 0\ \ \forall\ (i,j) \in I\!E\} \\ &= \max \{\ e^T x \ \mid\ \hat{X} = \begin{bmatrix} X & x \\ x^T & 1 \end{bmatrix},\ x = \operatorname{diag}(X),\ \ A_G \bullet X = 0,\ \ \operatorname{rank}(\hat{X}) = 1\} \\ &= \max \{\ J \bullet X \ \mid\ \hat{X} = \begin{bmatrix} X & x \\ x^T & 1 \end{bmatrix},\ x = \operatorname{diag}(X),\ \ A_G \bullet X = 0,\ \ \operatorname{rank}(\hat{X}) = 1\}^{1/2} \quad (4.5)\end{aligned}$$

where we introduce a binary variable x_i for each node $i \in V$. As defined on page 20, A_G denotes the adjacency matrix of G. The variables are constrained to be in $\{0,1\}$ indicating whether node i is chosen in the stable set or not. In place of the linear constraint in (4.3) we have bilinear edge constraints which state that for each edge $\{i,j\}$ it is not allowed to choose both nodes i and j for the stable set.
In the second equation we embed the binary vector x into a rank-one matrix $\hat{X}$. Note that the rank condition along with the constraint that the last entry of $\hat{X}$ is one implies that $\hat{X}$ has the form

$$\hat{X} = \begin{bmatrix} x \\ 1 \end{bmatrix} \begin{bmatrix} x \\ 1 \end{bmatrix}^T$$

for some vector x, and the constraint $x = \operatorname{diag}(X) = \operatorname{diag}(xx^T)$ implies that the components of x are from $\{0,1\}$. Thus, X is nonnegative, and $A_G \bullet X = 0$ implies $x_i x_j = 0\ \ \forall\ (i,j) \in I\!E$. The third equation is evident.

Since the objective functions $e^T x$ or $J \bullet X$ of the above formulation are linear, replacing the feasible set with its convex hull will not change the optimal value of the problems. Convex relaxations are one of the most powerful techniques for designing polynomial time approximation algorithms for NP-hard optimization. A further relaxation of the nonconvex rank constraints on $\hat{X}$ yields two semidefinite programs

$$\alpha(G) \le \max \{\ e^T x \ \mid\ \hat{X} = \begin{bmatrix} X & x \\ x^T & 1 \end{bmatrix} \succeq 0,\ \hat{X} \ge 0,\ x = \operatorname{diag}(X),\ \ A_G \bullet X = 0\} \qquad (4.6)$$

and

$$\alpha(G)^2 \le \max \{\ J \bullet X \ \mid\ \hat{X} = \begin{bmatrix} X & x \\ x^T & 1 \end{bmatrix} \succeq 0,\ \hat{X} \ge 0,\ x = \operatorname{diag}(X),\ \ A_G \bullet X = 0\} \qquad (4.7)$$

where the x-component of the optimal solutions of both problems approximates the optimal solution of the max-stable-set problem.
The Lovász number is defined as:

$$\theta(G) = \max \{\sum_{i,j} X_{ij} \mid \sum_i X_{ii} = 1,\ X_{ij} = 0\ \ \forall ij \in I\!E,\ X \succeq 0\}. \qquad (4.8)$$

Schrijver introduced the strengthening θ' of the Lovász θ-number by adding the nonnegativity constraint $X \geq 0$ as

$$\theta'(G) = \max \{ \; J \bullet X \;\; | \;\; X \succeq 0, \; X \geq 0, \;\; I \bullet X = 1, \;\; A_G \bullet X = 0 \}. \tag{4.9}$$

which provides an improved upper bound on $\alpha(G)$,

$$\alpha(G) \leq \theta'(G) \leq \theta(G).$$

Let x be the characteristic vector of a largest stable set of size k of a given graph. xx^T is a $\{0,1\}$-matrix whose $(ij)^{th}$ entry equals one if i and j are included in the stable set. The matrix $X = \frac{1}{x^T x} xx^T$ is a feasible solution of (4.9) with objective value k.
An equivalent problem of (4.9) is given by

$$\theta'(G) = \max \{ \; J \bullet X \;\; | \;\; X \succeq 0, \; X_{ij} \geq 0 \;\; \forall ij \notin \mathbb{E}, \; X_{ij} = 0 \;\; \forall ij \in \mathbb{E}, \; I \bullet X = 1 \}. \tag{4.10}$$

Let $\bar{G}$ be the complement graph of G. According to the Gerschgorin theorem, every eigenvalue λ of an arbitrary matrix $A \in \mathbb{R}^{n \times n}$ lies within at least one of the discs

$$D_i = \{ \; z \; | \;\; |z - a_{ii}| \leq \sum_{\substack{j=1 \\ j \neq i}}^{n} |a_{ij}| \},$$

Therefore, it is easy to see that $X = \frac{1}{n} I + \frac{1}{2n^2} A_{\bar{G}}$ is a *Slater point* for (4.10).

Remark 3 *The standard Slater condition requires the existence of a feasible point $\tilde{X}$ at which all inequality constraints are strictly satisfied. A weaker form of the Slater condition requires that all nonlinear (convex) inequality constraints are strictly satisfied. This weaker form of Slater condition for the primal and dual problem also guarantees the existence of a primal-dual optimal solution, see e.g. [15].*
Therefore, the definition of a Slater point will slightly vary from one problem to another. For instance although the standard Slater condition is satisfied by (4.10), the Slater condition is limited to its weaker form in (4.9).

Next we show that the semidefinite relaxation in (4.6) is at least as strong as the relaxation in (4.7), and the semidefinite relaxation in (4.7) is at least as strong as θ'.

Theorem 9 *Let X and x be any feasible solution for (4.7) (or (4.6)). Then*

$$e^T x \leq \sqrt{J \bullet X} \leq \theta'(G). \tag{4.11}$$

Proof. If the feasible solution X of (4.7) is zero, then $x = 0$ and the inequality (4.11) is true. Evidently, for any nonzero X such that x, X are feasible for (4.7), the matrix $X/(I \bullet X)$ is feasible for (4.9). Note that $I \bullet X = e^T x$ since $x = diag(X)$, therefore

$$\frac{J \bullet X}{I \bullet X} \leq \theta'(G) \Rightarrow J \bullet X \leq \theta'(G)(e^T x). \tag{4.12}$$

On the other hand,

$$\hat{X} \succeq 0 \Leftrightarrow X - xx^T \succeq 0 \Leftrightarrow z^T(X - xx^T)z \geq 0 \ \ \forall z \in \mathbb{R}^n$$

Let $z = e$, then

$$J \bullet X \geq (e^T x)^2 \tag{4.13}$$

proves the first inequality.
Note that the inequality (4.13) is strict for any interior feasible solution $\hat{X}$. The second inequality in (4.11) is an inevitable consequence of (4.12) and (4.13). □

The next theorem states that $\theta'(G)$ can be expressed as optimum value of the linear objective function $e^T x$ which is maximized over the feasible region of the semidefinite relaxation (4.6). It will also provide a proof to show that equality in (4.13) will hold for any optimal solution X and x. This theorem has been shown in [20] in case of omitting the nonnegativity constraint from both problems (4.9) and (4.6).

Theorem 10 *Let $\mathcal{M}$ be the set of feasible solutions to the problem (4.6),*

$$\mathcal{M} = \{ \ \hat{X} \ \mid \ \hat{X} = \begin{bmatrix} X & x \\ x^T & 1 \end{bmatrix} \succeq 0, \ \hat{X} \geq 0, \ \hat{X}_{ij} = 0 \ \ \forall ij \in I\!\!E, \ \hat{X}_{ii} = \hat{X}_{i,n+1} \ (i = 1, \dots, n)\},$$

then

$$\theta'(G) = \max \ \{ \ \hat{I} \bullet \hat{X} \ \mid \ \hat{X} \in \mathcal{M}\} \tag{4.14}$$

where $\hat{I} = \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix}$.

Proof. Let $\varphi(G)$ be the optimum value of $\hat{I} \bullet \hat{X}$ over the feasible set $\mathcal{M}$,

$$\varphi(G) = \max \ \{ \ \hat{I} \bullet \hat{X} \ \mid \ \hat{X} \in \mathcal{M}\}. \tag{4.15}$$

We know that $e^T x \leq \theta'(G)$ for all feasible solutions x to the problem (4.6) (by Theorem 9), Therefore $\varphi(G) \leq \theta'(G)$.
In order to show $\theta'(G) \leq \varphi(G)$, we follow the proof of Monique Laurent [20] modified to include inequality constraints. Let X be the optimal solution to the Lovász-Schrijver function θ' in (4.9). Since X is a positive semidefinite matrix as well as being nonnegative, there exists some matrix V such that $X = V^T V$. Assume that $v_1, \dots, v_n$ are the columns of V, then we have $X_{ij} = v_i^T v_j \geq 0$ for all $i, j = 1, \dots, n$. Accordingly,

$$\begin{aligned} \theta'(G) &= J \bullet X = e^T X e = e^T (v_1, \dots, v_n)^T (v_1, \dots, v_n) e \\ &= (\sum_{i=1}^n v_i)^T (\sum_{i=1}^n v_i) = \| \sum_{i=1}^n v_i \|^2 \end{aligned} \tag{4.16}$$

and

$$1 = I \bullet X = \sum_{i=1}^{n} \|v_i\|^2.$$

Set $R := \{i \in \{1,\ldots,n\} \mid v_i \neq 0\}$ and

$$\begin{aligned} u_{n+1} &:= \frac{1}{\sqrt{\theta'(G)}} \sum_{i=1}^{n} v_i = \frac{1}{\sqrt{\theta'(G)}} \sum_{i\in R} v_i, \\ u_i &:= \frac{v_i}{\|v_i\|} \quad \text{for } i \in R. \end{aligned}$$

Let $u_i, i \in \{1,\ldots,n\} \setminus R$ be an orthonormal set in the orthogonal complement of the space spanned by $\{v_i \mid i \in R\}$. Set

$$\hat{X} = DZD$$

where D denote a diagonal matrix with diagonal entries $u_{n+1}^T u_i, i = 1,\ldots,n,n+1$ and Z denotes a matrix such that $Z_{ij} = u_i^T u_j$ for $i,j = 1,\ldots,n,n+1$. Obviously, $\hat{X}$ is a positive semidefinite matrix.
By the definition of u_i's, it follows

$$u_i^T u_j = \begin{cases} 1 & i,j = 1,\ldots,n,n+1 \\ \geq 0 & \textit{else.} \end{cases}$$

Then, the feasibility of $\hat{X}$ with entries $\hat{X}_{ij} = (u_{n+1}^T u_i)(u_i^T u_j)(u_{n+1}^T u_j)$ for $i,j = 1,\ldots,n,n+1$ to the problem (4.15) is easy to determine. To complete the proof, we will show that $\theta'(G) \leq \sum_{i=1}^{n} \hat{X}_{ii}$ as follows: By the definition of u_{n+1} we have

$$\begin{aligned} \theta'(G) &= (\sum_{i=1}^{n} u_{n+1}^T v_i)^2 = (\sum_{i\in R} u_{n+1}^T v_i)^2 \\ &= (\sum_{i\in R} u_{n+1}^T u_i \|v_i\|)^2 \\ &\leq (\sum_{i\in R} (u_{n+1}^T u_i)^2)(\sum_{i\in R} \|v_i\|^2) \\ &= \sum_{i\in R} (u_{n+1}^T u_i)^2 = \sum_{i=1}^{n} \hat{X}_{i,n+1} \end{aligned}$$

where the inequality follows from the Cauchy-Schwarz inequality. The first equality follows from

$$\begin{aligned} \theta'(G) &= (\sum_{i=1}^{n} v_i)^T (\sum_{i=1}^{n} v_i) \\ &= (\sqrt{\theta'(G)} u_{n+1})^T (\sum_{i=1}^{n} v_i) \\ &= \sqrt{\theta'(G)} (\sum_{i=1}^{n} u_{n+1}^T v_i). \end{aligned}$$

□

Remark 4 *We point out that Theorems 9 and 10 are also true if the doubly nonnegative cone* $\{\ X \mid \ X \succeq 0,\ X \geq 0\}$ *is replaced with the semidefinite cone. Theorems 9 and 10 imply that (4.7) and (4.9) are equivalent.*

4.3 Strengthening of the Semidefinite Relaxation of the Maximum Stable Set Problem

The variable $\hat{X}$ in the doubly nonnegative relaxation (4.6) approximates matrices in the convex hull of rank-one matrices $\hat{x}\hat{x}^T$ where $\hat{x} = (x^T, 1)^T$ and $x \in \{0,1\}^{|V|}$. In particular, the inequality $\hat{X} \geq 0$ in (4.6) can be motivated by the fact that $\hat{x} \geq 0$ implies the inequality $\hat{x}\hat{x}^T \geq 0$. Similarly, the inequalities $0 \leq (x-e)(x-e)^T$ and $0 \leq x(e-x)^T$ motivated by $e - x \geq 0$ and $x \geq 0$ can be relaxed and lead us to the following constraints

$$X + J - xe^T - ex^T \geq 0, \qquad xe^T - X \geq 0 \tag{4.17}$$

that may be added to the formulation of (4.6) or (4.7).

4.4 Equivalent Representation of the Lovász-Schrijver Number

To compare different formulations of the θ'-number we derive another equivalent representation. The following problem is considered as a relaxation of (4.7)

$$\max \{\ J \bullet X \mid \ \hat{X} \succeq 0,\ \hat{X} \geq 0,\ I \bullet X = e^T x,\ A_G \bullet X = 0\}. \tag{4.18}$$

It is obvious that the relaxed constraint $I \bullet X = e^T x$ is satisfied by any solution X and x of (4.7) (and (4.6)).

Proposition 3 *The problem (4.18) is equivalent to (4.7).*

Proof. The following diagram shows the relation between the problems (4.7), (4.9) and (4.18), *i.e.* $A \to B$ says an optimal solution of (A) can be translated into a feasible solution of (B).

$$\begin{tikzcd}(4.7) \arrow[r, "clear"] & (4.18) \arrow[r, "(\star)"] & (4.9) \arrow[r, "(\star\star)"] & (4.7)\end{tikzcd}$$

To show $(\star)$, let X, x be an optimal solution of (4.18). Then $\frac{X}{e^T x}$ is a feasible solution of (4.9). For $(\star\star)$, see the proof of Theorem 10. In the second part of the proof it is shown how to construct a feasible solution to (4.7) from an optimal solution to (4.9). □

According to Theorems 9 and 10 and Proposition 3, the problems (4.7), Schrijver θ'-number (4.9) and (4.18) are mathematically equivalent and generate the bound θ' for the maximum stable set problem. The accuracy of the solutions of (4.7), (4.9) and (4.18) given by SDPNAL and the computation times are tabulated in Table 4.1. For each dimension n, 10 different random graphs with 60% edge density were constructed.

n	residual			running time		
	(4.9)	(4.7)	(4.18)	(4.9)	(4.7)	(4.18)
50	2.08e-6	4.44e-6	4.54e-6	15.6	29.1	28.5
100	9.13e-7	7.83e-7	5.78e-7	17.6	46.8	36.8
200	7.43e-7	5.54e-7	6.04e-7	11.2	22.7	22.2
400	8.74e-7	5.15e-7	4.83e-7	80.6	109.4	73.6

Table 4.1: Comparison between (4.9), (4.7) and (4.18)

The numerical results given in Table 4.1 show no significant difference in the norm of the residuals for the relaxations (4.9), (4.7) and (4.18). However, the computation time of the θ'-number problem (4.9) is less than the other two problems. It is somewhat surprising that the problems for $n = 50$ appear to be more difficult than for $n = 200$. For $n = 50$ the average residual (geometric mean) is higher and also the number of iterations used by SDPNAL is much higher, even the overall computation time is higher despite of the fact that for $n = 200$ the numer of variables is more than 10 times higher than for $n = 50$.

4.5 Compute the Residual

The rounding errors that appear in the process of solving an optimization program by any specialized software packages cause the numerical solution to differ from the exact solution. The subject of this section is to assess the accuracy of the approximate solution of the doubly nonnegative optimization program

$$\max \{\, C \bullet X \mid \; X \succeq 0,\; X \geq 0,\; \mathcal{A}(X) = b\}. \tag{4.19}$$

given by SDPNAL (or any other packages). In order to solve (4.19) with SDPNAL, we introduce a new variable $X_{(n)} \in \mathcal{S}^n$ into (4.19) and form the following equivalent problem

$$\max \{\, C \bullet X \mid \; X \succeq 0,\; X_{(n)} \geq 0,\; \mathcal{A}(X) = b,\; X - X_{(n)} = 0\} \tag{4.20}$$

and the dual problem of (4.20) is given as

$$\max \{\, b^T y \mid \; \mathcal{A}^*(y) + Z + S = C,\; S_{(n)} - Z = 0,\; S \succeq 0,\; S_{(n)} \geq 0\}. \tag{4.21}$$

Problem (4.21) is obtained by straight forward dualization, of course, the variable Z can be eliminated from (4.21). The optimal solutions of (4.20) and (4.21) satisfy the constraints as well as the conditions $XS = 0$ and $X_n \circ S_n = 0$ called the complementarity condition. Let (X, X_n, y, S, S_n) be the approximate solution given by SDPNAL. In order to indicate how accurate the given solution is, we use the following residual:

$$
\begin{aligned}
residual \ := \ & \Big(\frac{\|\mathcal{A}^*(y)+Z+S-C\|^2}{1+\|C\|^2}+\frac{\|S_{(n)}-Z\|^2}{1+\|Z\|^2+\|S_{(n)}\|^2}+\frac{\|S-\Pi_{\mathcal{S}^n_+}(S)\|^2}{1+\|S\|^2} \\
+ \ & \frac{\|S_{(n)}-\Pi_{\mathcal{N}^n}(S_{(n)})\|^2}{1+\|S_{(n)}\|^2}+\frac{\|XS\|^2}{1+\|X\|^2\|S\|^2}+\frac{\|X_{(n)}\circ S_{(n)}\|^2}{1+\|X_{(n)}\|^2\|S_{(n)}\|^2} \\
+ \ & \frac{\|X-\Pi_{\mathcal{S}^n_+(X)}\|^2}{1+\|X\|^2}+\frac{\|X_{(n)}-\Pi_{\mathcal{N}^n}(X_{(n)})\|^2}{1+\|X_{(n)}\|^2} \\
+ \ & \frac{\|\mathcal{A}(X)-b\|^2}{1+\|b\|^2}+\frac{\|X-X_{(n)}\|^2}{1+\|X\|^2+\|X_{(n)}\|^2}\Big)^{\frac{1}{2}}
\end{aligned}
$$

where the matrix norms are the Frobenius norm. This residual measures the violation of the necessary and sufficient optimality conditions.

4.6 Equivalent Transformations

To summarize the results obtained from previous discussions, the relaxations (4.9), (4.7), (4.6) and (4.18) are equivalent due to Theorems 9, 10 and Proposition 3. Both, the optimal solution of either the relaxations (4.7) or (4.6), can be transformed via the GW-procedure to a $\{0,1\}$-solution which is feasible to the maximum stable set problem. Likewise, the relaxations (4.6), (4.7) and (4.18) without the nonnegativity condition $\hat{X} \geq 0$ are equivalent to the θ-number (4.8).

4.7 Construction of a Valid Upper Bound for a Primal SDP

As discussed in Section 2.4, there are some properties in SDP that differ from those in LP. The strict complementary condition is one of these properties. It is well-known that there always exists a pair of primal and dual optimal solutions in LP satisfying the strict complementary condition. However, this is not necessarily true in SDP. Strict complementarity condition is essential in order to analyze the convergence rates of several algorithms for solving SDP (such as SDPNAL, SEDUMI). The lack of strict complementarity may cause numerical and theoretical difficulties [33]. It has been shown that primal nondegeneracy, dual nondegeneracy and strict complementarity hold generically in SDP [2], however for the SDP's considered in this thesis, this genericity condition often does not hold.

In what follows we discuss a technique for the computation of a valid upper bound for the optimal value of a generic primal SDP. The objective of this technique is to deal with the rounding errors which appear in the optimal solutions of the semidefinite programs. Consider the general form of SDPs (2.7) and (2.8). We will shortly show that the objective value $C \bullet \bar{X}$ where $\bar{X}$ is the (exact) solution of (2.7) can be bounded from above using an approximate

solution.
Let $\tilde{X}, \tilde{y}, \tilde{S}$ denote approximate solutions of (2.7) and (2.8). First, we generate a complementary solution: Define

$$Z = \frac{\tilde{X}}{\|\tilde{X}\|_F} - \frac{\tilde{S}}{\|\tilde{S}\|_F}.$$

Note that if $\tilde{X}$ and $\tilde{S}$ are exact solutions, then Z has the same eigenvectors as $\tilde{X}$ and $\tilde{S}$. The eigenvalue decomposition of Z implies $Z = UDU^T$. If we apply the orthogonal matrix U to the $\tilde{X}$ and $\tilde{S}$, the results will be nearly diagonal matrices, denoted by $\tilde{\Lambda}$ and $\tilde{\Sigma}$:

$$\tilde{\Lambda} = U^T \tilde{X} U, \quad \tilde{\Sigma} = U^T \tilde{S} U.$$

For an arbitrary square matrix A the Matlab command diag(diag(A)) outputs a diagonal matrix with the same diagonal elements as A. We set $\Lambda = diag(diag(\tilde{\Lambda}))$ and $\Sigma = diag(diag(\tilde{\Sigma}))$ and construct new solutions $\hat{X} = U\Lambda U^T$ and $\hat{S} = U\Sigma U^T$ where the diagonal matrices are projected to satisfy $\Lambda \geq 0, \Sigma \geq 0$ and $\Lambda\Sigma = 0$. We now define a new perturbed problem such that $\hat{X}$ and $\hat{S}$ are the exact solutions:

$$\min \{\hat{C} \bullet X \mid \mathcal{A}(X) = \hat{b}, \quad X \succeq 0\} \tag{4.22}$$

where $\hat{C} := \mathcal{A}^*(\hat{y}) + \hat{S} \approx C$ and $\hat{b} := \mathcal{A}(\hat{X})$.

We now show for generic cases that a small perturbation ΔC of the objective function C of (2.7) leads to an approximate solution $\hat{X}$ near $\bar{X}$. First we quote the following theorem from [9] without proof.

Theorem 11 *Let a linear map $\mathcal{A}$ and its conjugate $\mathcal{A}^*$ be defined by (2.1) and (2.2) with the assumption that the matrices A^i, $i = 1, \ldots, m$ are linearly independent, a vector $b \in \mathbb{R}^m$ and a matrix $C \in \mathcal{S}^n$ be the data of the primal and dual semidefinite programs (2.7) and (2.8). We assume that the problems (2.7) and (2.8) satisfy the Slater condition and the optimal solutions $\bar{X}$ of (2.7) and $\bar{y}, \bar{S}$ of (2.8) are unique and satisfy the strict complementary condition. If the data of (2.7) and (2.8) is perturbed by a sufficiently small perturbation Δb, ΔC and $\Delta\mathcal{A}$, then up to second order terms, the perturbations ΔX and Δy,ΔS of the solutions X and y,S at $\bar{X}$ and $\bar{y}$,$\bar{S}$ satisfy the nonsingular linear system:*

$$\begin{aligned} \mathcal{A}(\Delta X) &= \Delta b - \Delta\mathcal{A}(\bar{X}), \\ \mathcal{A}^*(\Delta y) + \Delta S &= \Delta C - \Delta\mathcal{A}^*(\bar{y}), \\ \Delta X \bar{S} + \bar{X} \Delta S &= 0. \end{aligned} \tag{4.23}$$

As a direct consequence of Theorem 11 we obtain the following result.

Theorem 12 *With the assumptions of Theorem 11, let*

$$\bar{X} := argmin \; \{ \; C \bullet X \mid \mathcal{A}(X) = b, \quad X \succeq 0\}$$

and

$$\hat{X} := argmin \ \{(C + \Delta C) \bullet X \mid \mathcal{A}(X) = b + \Delta b, \ X \succeq 0\}.$$

There exist a finite number $M > 0$ and $\epsilon_0 > 0$ such that for any number $\epsilon_0 > \epsilon > 0$ it follows

$$\|\hat{X} - \bar{X}\| < M\epsilon$$

if $\| \begin{bmatrix} \Delta b \\ \Delta C \end{bmatrix} \| \leq \epsilon.$

Consider the optimization (4.22). The Lagrangian function of (4.22) is defined as

$$\mathcal{L}(X, y, S) = \langle \hat{C}, X \rangle + (\mathcal{A}(X) - \hat{b})^T y + \langle X, S \rangle.$$

Then, the sensitivity of the optimal value of (4.22) with respect to small changes Δb of $\hat{b}$ is approximately

$$(D_{\hat{b}}\mathcal{L}(X, y, S))[\Delta b] = \hat{y}^T \Delta b \leq \|\hat{y}\| \cdot \|\Delta b\|, \tag{4.24}$$

Let $\bar{X}$ be an (exact) solution of (2.7) which is not typically achievable numerically. In order to find a valid upper bound for the optimal value $C \bullet \bar{X}$ we use the perturbed solution $\hat{X}$ as follows:

$$\langle C, \bar{X} - \hat{X} \rangle = \langle C - \hat{C}, \bar{X} - \hat{X} \rangle + \langle \hat{C}, \bar{X} - \hat{X} \rangle,$$

Theorem 12 and inequality (4.24) imply that $\langle C - \hat{C}, \bar{X} - \hat{X} \rangle \leq M\epsilon^2$ and $\langle \hat{C}, \bar{X} - \hat{X} \rangle \leq \|\hat{y}\| \cdot \|\Delta b\|$, respectively. Therefore, one has

$$\langle C, \bar{X} \rangle \leq \langle C, \hat{X} \rangle + \delta_{new}$$

where $\delta_{new} = M\epsilon^2 + \|\hat{y}\|\|\Delta b\|$. Unfortunately, it is computationally expensive to generate a reliable bound for δ.

Remark 5 *In order to analyze the sensitivity of semidefinite programming by Theorem 11, the Slater condition, the uniqueness of the solutions and the strict complementary condition must be satisfied. If Slater condition does not hold for the primal or dual problem, a tiny perturbation makes the problem infeasible. The strict complementary condition is necessary to have an invertible linear system (4.23) as assumed in Theorem 11. If this condition does not hold, we have $r + s < n$ where r and s are the number of positive eigenvalues of the primal and dual solution respectively and therefore (4.23) is not solvable.*

Chapter 5

A Branch and Bound Approach

Before we analyze the algorithm proposed in this thesis to solve the maximum stable set problem, we present a brief sketch of the method. The method we use starts with the Schrijver-θ' number (4.9) as the root of a branch and bound tree which provides an upper bound of $\alpha(G)$. The semidefinite relaxation is strengthened by adding some cutting planes in each step. The cuts are satisfied by any solution of the maximum stable set problem. The aim of the cut is to get more accurate approximate solutions to the maximum stable set problem. The main goal is to reduce the depth of the branch and bound tree starting from a basic relaxation by using semidefinite relaxations. We complement the semidefinite upper bound by applying the Goemans and Williamson [11] approach to generate approximate solutions to the maximum stable set. The randomized solutions from the Goemans-Williamson approach are corrected by a simple heuristic to generate a locally maximal stable set. As the optimal value $\alpha(G)$ is not known for an arbitrary graph G, the algorithm is terminated when the upper bound coincides with the lower bound on $\alpha(G)$.

5.1 A Branch and Bound Algorithm for the Maximum Stable Set

Our proposed algorithm is described in details in this section. In order to solve semidefinite programs incorporated in the algorithm we apply a Matlab up-to-date software called SDPNAL [35].

- Algorithm:
  The algorithm starts by solving the semidefinite relaxation (4.9). The Lovász-Schrijver θ'-number (4.9) serves as a valid upper bound of $\alpha(G)$ following from Theorem 9. As rounding errors which typically appear in the result of any approximation algorithm may cause inaccuracy in the final results, the bounding relaxation (4.9) may not give an accurate upper bound. Therefore, we aim at finding a valid upper bound for the

relaxation (4.9). Let X be a given approximate solution to (4.9). Since typically X is neither feasible nor optimal, we consider the dual problem and try to find a valid upper bound for (4.9).

1. Here we are producing a dual feasible solution close to the optimal solution of (4.9) through a small perturbation which will be discussed shortly.
   We rewrite (4.9) in the following form,
   $$\theta'(G) := \max \{ \; J \bullet X \;\; | \;\; E_{ij} \bullet X = 0 \;\; \forall ij \in I\!\!E, \;\;\; I \bullet X = 1, \;\; X \succeq 0, \;\; X \geq 0 \}. \tag{5.1}$$
   Here $E_{ij} = e_i e_j^T + e_j e_i^T$ for $i \neq j$ which stands for the symmetric matrix whose ij^{th} and ji^{th} entries equals 1 and all other entries are zero.
   The dual problem is
   $$\min \{ \; y_1 \; | \;\; y_1 I + \sum_{ij \in I\!\!E} y_{ij} E_{ij} - Z_1 - Z_2 = J, \;\;\; Z_1 \succeq 0, \;\; Z_2 \geq 0 \; \}. \tag{5.2}$$
   Let $\tilde{y}$ and $\tilde{Z}_1$ and $\tilde{Z}_2$ be approximate optimal dual solutions (given by SDPNAL). The aim is to find a feasible dual solution nearby the SDPNAL-solution. To this end, we add small perturbations to the given dual solution to fit all constraints. Set
   $$\begin{aligned} \hat{Z}_2 \;\; &:= \;\; \max\,(\tilde{Z}_2, 0), \;\; (componentwise) \\ \hat{Z}_1 \;\; &:= \;\; -J - \hat{Z}_2 + \tilde{y}_1 I + \sum_{ij \in I\!\!E} \tilde{y}_{ij} E_{ij} \end{aligned}$$
   Let λ_{min} be the smallest eigenvalue of $\hat{Z}_1$. Since the symmetric eigenvalue problem is perfectly well conditioned, the error in the computation of λ_{min} using Matlab is small. In our implementation we assume that this error is bounded by $100\|\hat{Z}_1\|_F \cdot \epsilon$. For details see Appendix A.
   If $\lambda_{min} < \|\hat{Z}_1\|_F \cdot 100 \cdot \epsilon$ where ϵ is the machine accuracy, a nearby feasible solution can be constructed by setting
   $$\bar{Z}_1 := \hat{Z}_1 + \delta I,$$
   $$\bar{y}_1 := \tilde{y}_1 + \delta \; and \;\; \bar{y}_{ij} := \tilde{y}_{ij}$$
   where $\delta = -\lambda_{min} + \|\hat{Z}_1\|_F \cdot 100 \cdot \epsilon$. The additional term $\|\hat{Z}_1\|_F \cdot 100 \cdot \epsilon$ is intended to eliminate the possible rounding errors during the computation of the eigenvalues of the matrix $\hat{Z}_1$. If $\lambda_{min} \geq \|\hat{Z}_1\|_F \cdot 100 \cdot \epsilon$, set
   $$\bar{Z}_1 := \hat{Z}_1, \qquad \bar{y} := \tilde{y}.$$
   A feasible solution of (5.2) is then given by $(\bar{Z}_1, \hat{Z}_2, \bar{y})$. By the strong duality it therefore holds that
   $$\theta'(G) \leq \bar{y}_1.$$

We point out that unlike the discussion in Section 4.7, finding an upper bound for (4.9) is possible because of the special structure of $\mathcal{A}$.

Remark 6 *A strictly feasible solution can be constructed to the dual problem of the Lovász-Schrijver number θ' in (4.9), e.g.*

$$y_1 = -2(n+1), \quad Z_1 = 2(n+1)I - J, \quad Z_2 = J.$$

According to the conic duality theorem, the relaxation (4.9) is solvable. Nevertheless SDPNAL is not always successful in solving (4.9). SDPNAL is assumed to be convergent under Slater condition, therefore applying SDPNAL to the problem (4.9) which does not have a strictly feasible point may cause a failure in the result. For this reason we intend looking at the problem below

$$\max \{ \; J \bullet X \;\; | \;\; X_{ij} = 0 \;\; \forall ij \in I\!\!E, \; I \bullet X = 1, \; X \in \hat{K} \} \tag{5.3}$$

which is equivalent to (4.9) and satisfies the weak form of the Slater condition. The cone $\hat{K}$ and its dual $\hat{K}^D$ are defined by Lemma 1 in Section 2.1.
In our numerical examples we observed that it may happen that SEDUMI successfully solves both (4.9) and (5.3), but SDPNAL solves neither of these two problems. This observation limits the usefulness of SDPNAL for the branch and bound approach proposed in this thesis.

- The feasible set of problem (4.7) typically contains X,x where X is not a rank-1-matrix. If $\beta \le \theta'(G)$ is a candidate for the optimal value in (4.5), the aim $X = xx^T$ and $x^T e = \beta$ motivates the equation $Xe = \beta x$ as an additional linear equation in (4.7) leading to a possibly sharper relaxation

$$\max \{ \; J \bullet X \;\; | \;\; \hat{X} = \begin{bmatrix} X & x \\ x^T & 1 \end{bmatrix} \succeq 0, \; \hat{X} \ge 0, \; x = \operatorname{diag}(X), \\ A_G \bullet X = 0, \quad Xe = \beta x \}. \tag{5.4}$$

The goal of forming (5.4) is to reduce the gap between $\alpha(G)$ and the solution of its relaxation. By adding some constraints to problem (4.7) which are satisfied in the problem (4.5), we aim at cutting the current optimal solution of (4.7). Lemma 10 below states that the problem (5.4) where $\beta = \theta'(G)$ does not present an improvement of (4.7).

Lemma 10 *Let X^*,x^* be an optimal solution of (4.7), then $X^* e = \theta'(G) x^*$.*

Proof. *The Schur complement of the matrix $\hat{X}^*$ implies $X^* - x^* {x^*}^T \succeq 0$. On the other hand, by Theorems 9 and 10 we have $e^T X^* e = (e^T x^*)^2$. Therefore $X^* - x^* {x^*}^T$ has the eigenvalue* 0 *corresponding to the eigenvector e,*

$$(X^* - x^* {x^*}^T) e = 0 \Longrightarrow X^* e = x^* ({x^*}^T e) = \theta' x^*.$$

□

The next lemma refers to the case that $\beta = \lfloor \theta'(G) \rfloor$ is chosen.

Lemma 11 *If problem (5.4) has a feasible solution, then the optimal value of (5.4) is less than or equal to* β^2.

Proof. Let X, x be a feasible solution of (5.4). For $\beta = \theta'(G)$, the claim of Lemma 11 follows from Lemma 10. Let $\beta < \theta'(G)$, the objective value of (5.4) is

$$J \bullet X = e^T X e = e^T(\beta x) = \beta e^T x \leq \beta \sqrt{J \bullet X}.$$

The last inequality follows from $J \bullet (X - xx^T) \geq 0$ and $J \bullet X \geq J \bullet (xx^T) = (e^T x)^2$. Therefore we have $J \bullet X \leq \beta^2$ for all feasible solutions of (5.4). □

Remark 7 *In our algorithm we set* $\beta = \lfloor \theta'(G) \rfloor$. *If* θ' *is not an integer number, we will have* $\beta < \theta'$. *In the algorithm* β *is reduced to* $\beta - 1$ *only if we can guarantee that* $\alpha(G) < \beta$. *If for a given* β *the obtained upper bound in (5.4) is not at least* β^2, *it follows that there is no stable set of size* β *and* β *can be reduced. However, numerical observations suggest that the relaxation (5.4) doesn't provide a stronger upper bound for* $\alpha(G)$ *than the Schrijver relaxation (4.9).*

Remark 8 *We were not able to find any numerical example for which the optimal value of (5.4) is less than* β^2.

- To complement the upper bound for $\alpha(G)$ obtained from a semidefinite relaxation we intend to obtain a valid lower bound on the stability number $\alpha(G)$ by generating a $\{0,1\}$-vector x and assign a variable to each vertex (say x_i for vertex i) as follows

$$x_i = \begin{cases} 1 & i \in S \\ 0 & \textit{else.} \end{cases}$$

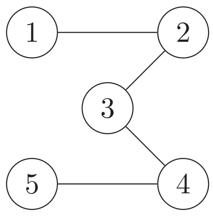
where S is a locally maximal stable set and x is called the characteristic vector of S. To this end we implement the randomized algorithm proposed by Goemans-Williamson [11] to generate many locally maximal stable sets:

1. Goemans and Williamson have proposed a randomized approach to generate a $\{-1,1\}$- rank-one-matrix from a positive semidefinite approximation with unit diagonal. The technique of Goemans and Williamson has been applied to the max-stable set problem as well, see *e.g.* Benson and Ye [3]. Below we outline a strategy that is equivalent to the approach by Benson and Ye. If there is a solution x and X to the problem (4.7)(or any other relaxations discussed later to provide an upper bound

on $\alpha(G)$), we set $\tilde{X} := 4X + J - 2xe^T - 2ex^T$. Note that $\operatorname{diag}(\tilde{X}) = e$ and $\tilde{X} \succeq 0$ since $X \succeq xx^T$. From the matrix $\tilde{X}$ we generate (by the Goemans-Williamson approach) random vectors $\tilde{x} \in \{-1, 1\}^n$ and transform these via $x = \frac{1}{2}(\tilde{x} + e)$ to random $\{0, 1\}$-vectors. We use the informal language saying that x contains vertices $i \in V$ if $x_i = 1$ and edges (i, j) if $x_i = x_j = 1$ *and* $(A_G)_{ij} = 1$, *i.e.* each random $\{0, 1\}$-vector x defines a sub-graph of G.

2. The random vectors x (obtained from (4.7)) are "reduced" by a greedy heuristic to make them characteristic vectors of stable sets:
   - First check whether x or $e - x$ contain less edges. In the latter case replace x with $e - x$.
   - Second, as long as there is an edge that is contained in x, identify a node in x that is adjacent to most nodes in x and delete this node.
   - Third, for each node v not in x, test whether adding v to x maintains the stable set property. If yes, add v to x.
   - Fourth, for each node v in x compute the set of all nodes $\tilde{v}$ that are not adjacent to any node in $x \setminus \{v\}$. If two or more such $\tilde{v}$ are not adjacent to each other then replace v with these nonadjacent $\tilde{v}$.
3. Store that largest stable set obtained this way. If the cardinality is less than β further branch-and-bound steps will be necessary.

Remark 9 *In the phase of deletion of nodes x which violate the stability of the set obtained by the GW-procedure above, there may exist more than one nodes with the same degree. The question arises as to which nodes to be removed first from the set in order to get a maximum stable set. As seen in the following graph,*



starting with node 2 or 4 we obtain the maximum stable set, while, this does not happen if we start with node 3.

Remark 10 *The fourth step of case 2 above is intended to possibly enlarge the size of the current stable set but it might be not useful in practice. Whether the fourth step is successful or not depends on the stable set x. Consider the example of Remark 9. Let the current stable set x have size 2.Then, the fourth step returns different results in case one of the sets $\{2, 4\}$ and $\{2, 5\}$ is selected as the stable set x. The application of this step to the latter set results in a maximum stable set, however this does not happen for the former. In our numerical* *experiments, it rarely ever happened that step 4 was used. In principle, the execution of step 4 is numerically cheap. When using Matlab, step 4 is more expensive since it is typically repeated many times and therefore more time-consuming in an interpreted language such as Matlab.*

Chapter 6

Branching Strategies for the Maximum Stable Set Problem

6.1 New Branching

Let $G = (\{1, \dots, n\}, I\!E)$ be a given graph and $G_i = (V_i, I\!E_i)$ denote a subgraph of G obtained by deleting node i from G and all adjacent nodes, *i.e.* with $V_i = \{1, \dots, n\} \backslash \{\{i\} \cup \{j \mid (i, j) \in I\!E\}\}$ and $I\!E_i = \{(i', j') \mid i', j' \in V_i, (i', j') \in I\!E\}$. Assume an upper bound β for the maximum stable set is known, *i.e.* $\alpha(G) \leq \beta$. Let $\tilde{x}$ be a $\{0, 1\}$-vector with $e^T \tilde{x} \geq \beta$. There are three possibilities:

1. There is a maximum stable set of size β in the graph $\tilde{G}$ where $\tilde{G}$ denotes the subgraph induced by the set of nodes i with $\tilde{x}_i = 1$, as defined on page 20.

2. For each maximum stable set S of G there exists at least one i with $\tilde{x}_i = 0$ and $i \in S$. This gives $p = n - e^T \tilde{x}$ $(\leq n - \beta)$ subcases. We assume (by reordering) that $\tilde{x}_1 = 0, \dots, \tilde{x}_p = 0$ and $\tilde{x}_{p+1}, \dots, \tilde{x}_n = 1$. For each subproblem i, $(1 \leq i \leq p)$, we assume that $\tilde{x}_i$ is in a maximum stable set. Delete node i and all nodes adjacent to i from G. This gives a smaller subgraph G_i. Let L_{α_i} and U_{α_i} denote the lower and upper bound on $\alpha(G_i)$, respectively obtained from a semidefinite relaxation and the Goemans-Williamson heuristics. If there is one i, $1 \leq i \leq n$, such that for the smaller subgraph G_i we obtain $L_{\alpha_i} = U_{\alpha_i} = \beta - 1$, we are done, *i.e.* $Ł_{\alpha_i}$ and U_{α_i} provide a proof for $\alpha(G) = \beta$. If $U_{\alpha_i} < \beta - 1$ for all i with $\tilde{x}_i = 0$, β must be reduced by one. In this case, if for one i we have $L_{\alpha_i} = U_{\alpha_i} = \beta - 2$, then $\alpha(G) = L_{\alpha_i} + 1 = U_{\alpha_i} + 1 = \beta - 1$. Else, for those G_i where $U_{\alpha_i} \geq \beta - 1$, the same algorithm must be applied, *i.e.* a new vector $\tilde{x}$ must be defined for G_i and the algorithm must be repeated.

3. Any maximum stable set of G has cardinality "$\leq \beta - 1$".

In our algorithm, the semidefinite relaxation problem (4.6) (or (4.7)) provides an upper bound on $\alpha(G_i)$ and a lower bound is provided by the Goemans and Williamson approach.

6.1.1 Possible Strategies to Choose $\tilde{x}$

1. Let $G = (V, E)$ be an arbitrary graph and $S = \{i_1, \dots, i_k\} \subset V$ be a locally maximal stable set of size k. For $i_l \in S$, $l = 1, \dots, k$ let $N(i_l)$ be the set neighbors of i_l. In the following we intend to construct a clique covering of S. For each $i_l \in S$, $l = 1, \dots, k$ we identify a locally maximum clique $C(i_l)$ in $(N(i_l) \setminus \bigcup_{j=1}^{l-1} C(i_j)) \cup \{i_l\}$. By construction it follows that $i_l \in C(i_l)$ and $C(i_l) \cap C(i_k) = \varnothing$ for $l \neq k$ and at most one element of each $C(i_l)$ can be contained in a stable set. Set
$$\tilde{x}_j = \begin{cases} 1 & \text{if } j \in \bigcup_{i_l \in S} C(i_l) \\ 0 & \textit{else.} \end{cases}$$
Here, the set $\{i \mid \tilde{x}_i = 1\}$ is not a stable set in general, but if the maximum stable set has cardinality larger than $|S|$ it must contain some j with $\tilde{x}_j = 0$. If one can not find any stable set of size larger than $|S|$, the set S is an optimal solution.

2. The strategy mentioned in Step 2 of Section 6.1 to reduce β contains p subproblems (and p could be large). Let $nv(i)$ denote the number of nodes adjacent to the node i. Each subproblem $i \in \{1, \dots, p\}$ seeking for an upper bound on $\alpha(G_i)$ has $n - nv(i) - 1$ binary variables. We seek a vector $\tilde{x} \in \{0,1\}^n$ with at least β components of value one such that the dimension of each subproblem i becomes as small as possible. To this end, sort the neighbor vector nv in decreasing order, called snv. The nodes corresponding to the last β entries of snv are set to one and the rest zero. For this approach, this choice of $\tilde{x}$ reduces the dimension of the subproblems in the next branch and bound level to the smallest possible dimension.

3. There is an alternative to choose the vector $\tilde{x}$. Let $\tilde{x} \in \{0,1\}^n$ be a locally maximum stable set of the graph G obtained by the GW-rounding algorithm. If $e^T\tilde{x} < \beta$, among the nodes i with $\tilde{x}_i = 0$ we set those nodes to one which have the least number of neighbors until we get $e^T\tilde{x} \geq \beta$. Here, the approach of the previous paragraph is slightly changed by including a locally maximum stable set among the nodes i with $\tilde{x}_i = 1$.

Remark 11 *Step 2 of Section 6.1 can be generalized aiming at reduction of β by two or more. Let $\tilde{G}$ denote the graph with nodes i with $\tilde{x}_i = 1$. If for all subgraphs G_i, $U_{\alpha_i} < \beta - k$ where $k \geq 2$, β will be reduced by k if also the maximum stable set of $\tilde{G}$ has size " $\leq \beta - k$ ".*

6.2 An Extension of the New Branching

In what follows, we make an extension of the case 1 of the Section 6.1. Let a maximum stable set be contained in the set $\{i \mid \tilde{x}_i = 1\} =: \tilde{I}$ where $|\tilde{I}| = \beta$. We discussed above how to reduce β to $\beta - 1$. In this section the case will be argued where β can be reduced by 2. We claim that if all edges in $\tilde{I}$ are pairwise adjacent, then there is a stable set in $\tilde{I}$ of size either $\beta - 1$ or $\beta - 2$. In order to support this claim, all the possibilities where any 2 edges in an arbitrary graph are adjacent are listed in the following figure.

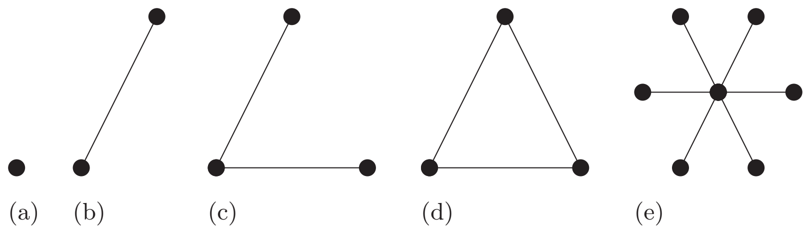


Figure 6.1: Example graphs

Here, case (e) refers to the situation where there are more than two nodes adjacent to the node in the middle. As we can see from the picture, the graphs 1.b, 1.c and 1.e turn into a stable set by eliminating one node. Only the graph 1.d has a stable set of size one and in order to form a stable set one needs to remove two nodes from the graph. Therefore, the cardinality of $\tilde{I}$ can be reduced by one or two if all edges in $\tilde{I}$ are as in one of the graphs 1.a-1.e.

6.3 Alternative Strategy for the Branching Step

If there is a gap between the solution provided by the relaxation (4.7) and the solution given by the GW-rounding algorithm, we will incorporate new constraints to the relaxations in order to reduce the gap. This issue will be discussed in this section.

The relaxation (4.7) can be strengthened by adding additional constraints which are satisfied by the rank-one-matrix solution of the maximum stable set problem. It is obvious that for any vector $v \in \mathbb{R}^n$ the constraint $Xv = \gamma x$ with $\gamma = v^T x$ is satisfied by any rank one solution of the stable set problem. The new relaxation including the constraint $Xv = \gamma x$ is supposed to provide a better approximation of the stable set problem for some vectors v. For this, the strategy is to define a vector v in a way that the constraint $Xv = \gamma x$ separates the current optimal solution X^*, x^* of (4.7) from the convex hull of the set of feasible solutions of the new relaxation and provides a possibly deep cut and therefore a stronger relaxation. The method of choosing v will be explained in more detail below.
It follows two conflicting goals. On the one side, v is to be chosen such that the equation

$Xv = \gamma x$ is violated for any $\gamma \in \mathbb{R}$, more precisely, such that

$$\min_{\gamma \in \mathbb{R}} \frac{\|Xv - \gamma x\|}{\|v\|}$$

is large. On the other side, v is to be chosen such that there are only few possible values of γ to be considered. In order to reduce the number of different values of γ that need to be considered in a branch and bound approach, the vector v maximizing $\frac{\|Xv-\gamma x\|}{\|v\|}$ is rounded to a nearby integer vector. More precisely, after finding an appropriate vector v, called $\hat{v}$ with $\|\hat{v}\|_\infty = 1$, round $\hat{v}$ to the nearest vector v in $\{-1, 0, 1\}^n$. Let $B = X - xx^T$ where X, x is the solution of (4.7). Test whether $\|Bv\|_2$ satisfies ,

$$\|Bv\|_2 \geq \eta.\lambda_{max}(B).\|v\|_2. \tag{6.1}$$

with some fixed parameter $\eta \in (0, 1)$, *e.g.* $\eta = 0.1$. If not, modify v.
Let the number of -1 and 1 in v be $\hat{t}$ and t, respectively. For

$$\gamma = -\hat{t}, -\hat{t} + 1, \ldots, t - 1, t, \tag{6.2}$$

solve the semidefinite relaxation

$$\max \; \{ \; e^T x \;\; | \;\; \hat{X} = \begin{bmatrix} X & x \\ x^T & 1 \end{bmatrix} \succeq 0, \;\; \hat{X} \geq 0, \;\; x = \operatorname{diag}(X), \quad A_G \bullet X = 0, \quad Xv = \gamma x\}. \tag{6.3}$$

If (6.3) does not have a solution for some value of γ, then this value of γ can be dropped. If for each value of γ the optimal value of (6.3) is less than β, then β can be reduced to $(\beta - 1)$. If (6.3) has a solution, then try to identify a rank-one-solution from it *e.g.* via the Goemans-Williamson approach of Section 5.1. If necessary repeat this branching by adding further vectors $\tilde{v}$ with associated values $\tilde{\gamma}$.

Remark 12 *According to Theorems 9 and 10, the equality $J \bullet X = (e^T x)^2 = (\theta')^2$ holds for any optimal solution of the relaxation (4.7). However, this equality is not true in general. It follows from the conic constraint $\hat{X} \succeq 0$ that $X - xx^T \succeq 0$ and therefore $J \bullet X \geq (e^T x)^2$. Therefore, we use the term $e^T x$ as the objective value of (6.3) to achieve a possibly stronger relaxation.*

Remark 13 *Let X, x be a solution of (4.7). The matrix $B := X - xx^T$ is positive semidefinite. If it were zero, the max-stable set problem would be solved. Since B is positive semidefinite but not positive definite, there exist one or more eigenvectors v to the eigenvalue 0. Let v be an eigenvector corresponding to the eigenvalue zero. Since the (current) optimal solution X, x of (4.7) satisfy the new constraint $Xv = \gamma x$ with $\gamma = v^T x$, this choice of v doesn't make any improvement in the approximation. Let v be an eigenvector corresponding to the maximum eigenvalue of B denoted by λ_{max}. Clearly, $\|Xv - \gamma x\| = \lambda_{max}\|v\| \neq 0$ where $\gamma = v^T x$. This motivates us to use the inequality (6.1) as a criterion to test v after the rounding procedure.*

In what follows, it will be discussed how to generate a vector v such that the current solution X, x violates the additional constraint $Xv = \gamma x$ for any γ. To this end, we approach the following problem in three different ways:

$$\max_{v\in\mathbb{R}^n,\ \|v\|_2=1}\ \min_{t\in\mathbb{R}}\ \|Xv - tx\|_2^2. \tag{6.4}$$

1) **First Version of Generating a Vector v**

   In order to solve (6.4), let

$$\begin{aligned} f(t,v) &= \|Xv - tx\|_2^2, \\ &= v^T X^2 v + t^2 x^T x - 2t x^T X v. \end{aligned}$$

   Since $f(t,v)$ is a convex quadratic function, the first order condition for minimizing with respect to t namely $\frac{\partial f(t,v)}{\partial t} = 0$ implies that

$$t = \frac{x^T X v}{x^T x}$$

   is the solution of the minimization problem. Replacing the obtained value t in (6.4) and reformulation of the fraction $\frac{(x^T X v)^2}{x^T x} := \frac{(v^T X^T x)(x^T X v)}{x^T x}$ leads to the following equivalent problem:

$$\max_{v\in\mathbb{R}^n,\ \|v\|_2=1}\ v^T (X^2 - \frac{Xx(Xx)^T}{x^T x}) v. \tag{6.5}$$

   The optimal solution vector v is known as the unit-norm eigenvector corresponding to the maximum eigenvalue of $(X^2 - \frac{Xx(Xx)^T}{x^T x})$. This vector v and the two following alternatives for generating a vector v are compared with each other in Section 8.2.

2) **Second Version of Generating a Vector v**

   The maximization (6.5) can be represented in the following equivalent form:

$$\max_{v\in\mathbb{R}^n,\ \|v\|_2=\sqrt{n}}\ v^T (X^2 - \frac{Xx(Xx)^T}{x^T x}) v. \tag{6.6}$$

   For convenience of notation let $Z = X^2 - \frac{Xx(Xx)^T}{x^T x}$. Clearly, any $v \in \{+1, -1\}^n$ is a feasible solution to (6.6). The GW-rounding procedure is well-known to produce a $\{-1, +1\}$-vector. This motivates us to relax the optimization (6.6) to

$$\max\ \{V \bullet Z \mid\ V \succeq 0,\ \operatorname{diag}(V) = e\ \}. \tag{6.7}$$

   The GW-method with different rounding procedure can be applied to the solution V of (6.7) in order to generate a vector $v \in \{-1, 0, +1\}$. Here, the rounding procedure is modified as follows: Since each component i with $v_i \neq 0$ induces an additional value of γ that needs to be considered in (6.2), it is aimed to round many components of v to zero. To this end, let $k := \max\ |Lu|_i$ be the maximum component of Lu. Let $\eta \in (0,1)$ be a fixed parameter, if $|Lu|_i > \eta k$, set $v_i := sign(Lu)_i$; else set $v_i = 0$.

3) **Third Alternative Version of Generating a Vector v**

Note that problem (6.4) serves as a heuristics to generate a deep cut. The optimal solution of (6.4) must be perturbed to allow for a suitable branching strategy. Thus, finding the exact solution of (6.4) is not crucial. If the rounding procedure does not provide a vector v satisfying (6.1) we consider a simplified version of (6.4) obtained by adding the constraint $x^TXv = 0$ to (6.5). The motivation for this constraint is that v is to be chosen such that Xv is not a multiple of x at the optimal solution X, x of (6.3) (unless X is a rank-one-matrix $X = xx^T$ in which case the algorithm stops). To eliminate the possibility that $Xv = tx$ we require $\|v\|_2 = 1$ and $x^TXv = 0$. So the modified version of (6.5) can be restated as follows:

$$\max \{ \; v^TX^2v \mid \; x^TXv = 0, \;\; \|v\|_2 = 1\} \tag{6.8}$$

The eigenvalue decomposition of X yields $X = UDU^T$ and then $X^2 = UD^2U^T$. Thus, (6.8) is equivalent to

$$\max\{v^TUD^2U^Tv \mid \; x^TXUU^Tv = 0, \; \|v\|_2 = 1\}.$$

Set

$$\tilde{v} := U^Tv, \quad \tilde{r} := U^TXx,$$

therefore the problem (6.8) has been translated into the following problem which has an appropriate structure to deal with,

$$\max \; \{\tilde{v}^TD^2\tilde{v} \;\mid \tilde{v}^T\tilde{r} = 0, \;\; \|\tilde{v}\|_2 = 1\}. \tag{6.9}$$

The next theorem will provide a solution for (6.9).

Theorem 13 *Any normalized eigenvector to the largest eigenvalue of $\tilde{D}$ where*

$$\tilde{D} := (I - \frac{\tilde{r}\tilde{r}^T}{\|\tilde{r}\|_2^2})D^2(I - \frac{\tilde{r}\tilde{r}^T}{\|\tilde{r}\|_2^2}),$$

is an optimal solution of (6.9).

Proof. For the sake of convenience denote

$$\Pi := (I - \frac{\tilde{r}\tilde{r}^T}{\|\tilde{r}\|_2}), \; \tilde{D} := \Pi D^2\Pi.$$

Here, Π is the orthogonal projection onto $\{\tilde{r}^\perp\}$. Let $\hat{v}$ be a unit eigenvector corresponding to the largest eigenvalue of $\tilde{D}$. Note that $\tilde{r}$ is an eigenvector corresponding to the eigenvalue zero of the matrix $\tilde{D}$ as $\tilde{D}\tilde{r} = 0$. Inasmuch as the eigenvectors of different eigenvalues of symmetric matrices are orthogonal, it then follows $\hat{v}^T\tilde{r} = 0$. Therefore

$$\begin{aligned} \lambda_{max}(\tilde{D}) &= \hat{v}^T\tilde{D}\hat{v} \\ &= \hat{v}^T\Pi D^2\Pi\hat{v} = \hat{v}^TD^2\hat{v}, \end{aligned} \tag{6.10}$$

On the other hand, the Raleigh-Ritz theorem states that for the symmetric matrix $\tilde{D}$ one has

$$\begin{aligned} \lambda_{max}(\tilde{D}) &= \max_{z\in\mathbb{R}^n,\ \|z\|_2=1} z^T\tilde{D}z \\ &= \max_{z\in\mathbb{R}^n,\ \|z\|_2=1} z^T \mathit{\Pi} D^2 \mathit{\Pi} z \\ &= \max_{z\in(\tilde{r})^\perp,\ \|z\|_2\le 1} z^T D^2 z \\ &= \max_{z\in(\tilde{r})^\perp,\ \|z\|_2=1} z^T D^2 z, \end{aligned} \tag{6.11}$$

where the last equality follows form the fact that $D^2 \succeq 0$. In order to show the third equality, assume that $\hat{z}$ is an optimal solution of

$$\max_{z\in\mathbb{R}^n,\ \|z\|_2=1} z^T \mathit{\Pi} D^2 \mathit{\Pi} z.$$

According to Theorem 7, $\hat{z}$ can be decomposed to $\hat{z} = \hat{z}_1 + \hat{z}_2$ such that $\hat{z}_1 \in \{\lambda\tilde{r}\}$ and $\hat{z}_2 \in \{\tilde{r}^\perp\}$. As $\hat{z}_1 \perp \hat{z}_2$, then $\|\hat{z}_1\|^2 + \|\hat{z}_2\|^2 = 1$ and the optimal objective value is

$$(\hat{z}_1+\hat{z}_2)^T \mathit{\Pi} D^2 \mathit{\Pi} (\hat{z}_1+\hat{z}_2) = \hat{z}_2^T D^2 \hat{z}_2.$$

where $\|\hat{z}_2\|_2 \le 1$. The equalities (6.10) and (6.11) imply

$$\hat{v}^T D^2 \hat{v} \ge \tilde{v}^T D^2 \tilde{v} \ \ \forall \tilde{v}\ s.t.\ \tilde{v} \perp \tilde{r},\ \|\tilde{v}\|_2 = 1.$$

which completes the proof.

□

In order to apply SDPNAL, problem (6.3) is represented as follows:

- $$x = \operatorname{diag}(X) \Leftrightarrow \hat{X}_{i,i} - \hat{X}_{i,n+1} = 0 \ \ \forall i = 1,\dots,n.$$

  For each $i \in \{1,\dots,n\}$ we define a symmetric $(n+1)\times(n+1)$-matrix $S^{(i)}$ as

  $$(S^{(i)})_{i,j} = \begin{cases} 2 & i=j \\ -1 & j=n+1 \text{ or } i=n+1 \\ 0 & else. \end{cases} \tag{6.12}$$

  Therefore,

  $$x = \operatorname{diag}(X) \Leftrightarrow \quad S^{(i)} \bullet \hat{X} = 0 \ \ \forall i = 1,\dots,n.$$

  Note that S^i is very sparse with only 3 nonzero entries and note that this form of sparsity is fully exploited by SDPNAL.

- Add a zero column and a zero row to the right and bottom of the matrix A_G. The results are denoted by

$$\hat{A}_G = \begin{bmatrix} A_G & 0 \\ 0 & 0 \end{bmatrix}. \tag{6.13}$$

Then

$$A_G \bullet X = 0 \Leftrightarrow \; \hat{A}_G \bullet \hat{X} = 0,$$

- We define a new matrix U whose $(n+1, n+1)^{th}$ entry equals 1 and the rest is zero and add constraint $U \bullet \hat{X} = 1$ to fix $\hat{X}_{n+1,n+1} = 1$.

- To translate the constraint $Xv = \gamma x$, we have

$$Xv = \gamma x \Leftrightarrow N^i \bullet \hat{X} = 0 \;\; \forall i = 1, \ldots, n.$$

where

$$N^{(i)} = \begin{bmatrix} 0 & \cdots & 0 & v_1 & 0 & \cdots & 0 & 0 \\ & \vdots & & \vdots & & \vdots & & \vdots \\ 0 & \cdots & 0 & v_{i-1} & 0 & \cdots & 0 & 0 \\ v_1 & \cdots & v_{i-1} & 2v_i & v_{i+1} & \cdots & v_n & -\gamma \\ 0 & \cdots & 0 & v_{i+1} & 0 & \cdots & 0 & 0 \\ & \vdots & & \vdots & & \vdots & & \vdots \\ 0 & \cdots & 0 & v_n & 0 & \cdots & 0 & 0 \\ 0 & \cdots & 0 & -\gamma & 0 & \cdots & 0 & 0 \end{bmatrix} \tag{6.14}$$

is a symmetric matrix of order $n+1$. Here only the i^{th} row and the i^{th} column differ from the zero matrix.

6.4 Check the Nontrivial Infeasibility of the Relaxation (6.3) of the Branching Step

We anticipate that problem (6.3) may not be feasible for some possible values of $\gamma \in \{-\hat{t}, \ldots, t\}$. In what follows, we address how to prove infeasibility in the presence of rounding errors.

Below we will define a new problem which is specialized in providing a proof of infeasibility of the problem (6.3). Since the existence of a KKT-solution to a semidefinite optimization problem in the combinatorial algorithms such as SDPNAL relies on the strict feasibility condition, we intend to define a problem satisfying the strict feasibility condition. To this end, let X^*, x^* like

$$\begin{aligned} X^* &:= \frac{1}{2n} I + \frac{1}{4n^2} \mathcal{A}_{\bar{G}} \\ x^* &:= \frac{1}{2n} e, \end{aligned}$$

be a strictly feasible solution of problem (4.7). Denote $res := X^*v - \gamma x^*$ where the vector v is part of data generated as in cases 1, 2 and 3 of Section 6.3 and not a variable. Consider the optimization

$$\begin{aligned} min\ \{\lambda \mid\ \hat{X} \succeq 0,\ \hat{X} \geq 0,\ x = \operatorname{diag}(X),\ A_G \bullet X = 0,\ Xv = \gamma x + \lambda res, \\ e^T x \geq 1,\ \lambda \geq 0\}. \end{aligned} \tag{6.15}$$

It is obvious that the problem has a strictly feasible solution for $\lambda = 1$, namely X^*, x^*. Hence, (6.15) also has a finite optimal value $\lambda^{opt} \geq 0$. By the assumption that a maximum stable set of a graph G has the size of at least one, the constraint $e^T x \geq 1$ is satisfied by any rank-one solution to the maximum stable set problem. Since, the linear system of the first line in (6.15) is homogeneous, in order to avoid the trivial zero solution the constraint $e^T x \geq 1$ is added. If the optimal value of the problem (6.15) is strictly positive, the relaxation (6.3) must be infeasible.

In the numerical examples in Section 8.2, SDPNAL always generated a solution that was nearly dual feasible up to the typical truncation error. Thus, while problem (6.15) may be of theoretical interest, it was not used in our examples in Section 8.2.

Chapter 7

Comparison, Zero-One vs. Plus-Minus-One Formulation

Let x be a vector in $\{0,1\}^n$. The transformation $z := 2x - e$ translates a zero-one programming problem to a minus-one-plus-one problem through the following transformations:

$$Z \quad := \quad zz^T = (2x-e)(2x-e)^T = 4X - 2ex^T - 2xe^T + J \tag{7.1}$$

or

$$X \quad := \quad xx^T = \frac{1}{4}(z+e)(z+e)^T = \frac{1}{4}(Z + ze^T + ez^T + J). \tag{7.2}$$

Therefore, the relaxation (4.6) can be presented in another equivalent form as

$$\begin{array}{ll} \max & \frac{1}{2}e^T(z+e) \\ \text{s.t.} & \operatorname{diag}(Z) = e \\ & A_G \bullet (Z + ez^T + ze^T + J) = 0 \\ & Z + ez^T + ze^T + J \geq 0 \\ & \hat{Z} = \begin{bmatrix} Z & z \\ z^T & 1 \end{bmatrix} \succeq 0 \end{array} \tag{7.3}$$

which forms the minus-one-plus-one relaxation of the maximum stable set problem. The Benson and Ye approach [3] introduced in the next section applies a ± 1-formulation in order to find a maximum stable set of a given graph.

7.1 Benson and Ye Approach

In what follows we briefly describe the Benson and Ye approach [3] to address the maximum stable set problem, shortly MSS. First, in the following lemma we will show that adding an additional vertex $n+1$ does not worsen the semidefinite approximation θ of the MSS.

Let G_{new} be the graph obtained by adding an additional vertex $n+1$ not connected to the other vertices of a given graph G. Obviously, the vertex $n+1$ is contained in any maximum stable set. For the sake of simplicity, the notation θ refers to $\theta(G)$ and θ_{new} to $\theta(G_{new})$ where $\theta(G_{new})$ is the Lovász θ-number (4.8) of the graph G_{new}.

Lemma 12 *With the above notation it holds $\theta_{new} = \theta + 1$.*

Proof. First, we will show that

$$1 + \theta \leq \theta_{new}. \tag{7.4}$$

Let $\hat{X} = \begin{bmatrix} X & x \\ x^T & 1 \end{bmatrix}$ be an optimal solution of

$$\max \left\{ e^T x \;\middle|\; \begin{bmatrix} X & x \\ x^T & 1 \end{bmatrix} \succeq 0,\ x = \operatorname{diag}(X),\ X_{ij} = 0\ \ (i,j) \in I\!\!E \right\} \tag{7.5}$$

which is equivalent to θ (see Section 4.6). Note that the matrix $\hat{X}$ is of dimension $(n{+}1)\times(n{+}1)$. Clearly, $\frac{\hat{X}}{I\bullet\hat{X}}$ is a feasible solution of θ_{new} with the objective value

$$J \bullet \left(\frac{\hat{X}}{I \bullet \hat{X}}\right) \leq \theta_{new}.$$

On the other hand,

$$\begin{aligned} \frac{J \bullet \hat{X}}{I \bullet \hat{X}} &= \frac{1 + J \bullet X + 2e^T x}{1 + I \bullet X} = \frac{1 + (I \bullet X)^2 + 2(I \bullet X)}{1 + I \bullet X} \\ &= \frac{(1 + I \bullet X)^2}{1 + I \bullet X} = 1 + I \bullet X = 1 + \theta. \end{aligned}$$

In the second equation above, Theorems 9 and 10 are used to conclude that $J\bullet X = (e^T x)^2 = \theta^2$. This shows (7.4). Second, to show the inequality "$\theta + 1 \geq \theta_{new}$" we assume that $\hat{Y} = \begin{bmatrix} Y & y \\ y^T & \zeta \end{bmatrix}$ is an optimal solution of θ_{new} defined via (4.8) for G_{new}. It is obvious that $\frac{Y}{I\bullet Y}$ is a feasible solution for θ with objective value $\frac{J\bullet Y}{I\bullet Y} \leq \theta$.

Note that $\theta_{new} = J \bullet Y + \zeta + 2e^T y$ and $1 = \zeta + I \bullet Y = I \bullet \hat{Y}$. Therefore

$$\theta_{new} - \zeta - 2e^T y = J \bullet Y \leq \theta(I \bullet Y) = \theta(1 - \zeta). \tag{7.6}$$

In what follows we will show that

$$2e^T y = \zeta\theta + 1 - \zeta. \tag{7.7}$$

Note that substituting $2e^T y$ in (7.6) will end the proof.

To prove (7.7), since

$$J \bullet Y \leq \theta(1 - \zeta),$$

therefore

$$1-\zeta-J\bullet Y\geq 1-\zeta-\theta(1-\zeta). \tag{7.8}$$

On the other hand

$$1-(\zeta+J\bullet Y)=1-(\theta_{new}-2e^Ty)=1-\theta_{new}+2e^Ty\leq -\theta+2e^Ty. \tag{7.9}$$

The inequality in (7.9) follows from (7.4). The inequalities (7.8) and (7.9) imply that

$$-\theta+2e^Ty\geq 1-\zeta-J\bullet Y\geq 1-\zeta-\theta+\zeta\theta.$$

Therefore,

$$2e^Ty\geq 1-\zeta+\zeta\theta. \tag{7.10}$$

Furthermore, since $\hat{Y}\succeq 0$, the Schur complement implies that $Y-\frac{1}{\zeta}yy^T\succeq 0$. Then, $e^TYe\geq \frac{1}{\zeta}(e^Ty)^2$. Then, we have

$$(2e^Ty)^2=4(e^Ty)^2\leq 4\zeta(J\bullet Y)\leq 4\zeta\theta(1-\zeta)\leq(\zeta\theta+1-\zeta)^2. \tag{7.11}$$

where the second inequality follows from (7.6) and the last from $1-\zeta\geq 0$. In conclusion, the inequalities (7.10) and (7.11) complete the proof. □

Benson and Ye reformulate the MSS problem with ±1-variables. Each node of a given graph is assigned the value 1 or -1. Then a cut problem analogous to the Goemans-Williamson max-cut problem is defined such that the target of the optimal cut is to divide the nodes into two sets of nonadjacent and adjacent nodes. In order to detect the nonadjacent nodes, an artificial node, namely x_{n+1}, with no edge connecting to the other vertices is added to the graph. As the node x_{n+1} must be in any maximum stable set, the constraint

$$|x_i+x_j+x_{n+1}|=1,\ (x_i,x_j)\in I\!E$$

guarantees that nodes x_i, x_j and x_{n+1} can not be sided together in the optimal cut. Therefore, the MSS problem can be stated as follows

$$\begin{aligned} \max\quad & \frac{1}{2}(\sum_{i=1}^n x_i^2+x_{n+1}x_i)\\ \text{s.t.}\quad & x\in\{1,-1\}^n\\ & |x_i+x_j+x_{n+1}|=1,\ (x_i,x_j)\in I\!E. \end{aligned} \tag{7.12}$$

The objective function $\frac{1}{2}(\sum_{i=1}^n x_i(x_i+x_{n+1}))$ is intended for finding a set $S\subset V$ of nodes x_i with maximum cardinality in the same set as x_{n+1}. The following optimization problem refers to the semidefinite relaxation of problem (7.12)

$$\begin{array}{ll} \max & \begin{bmatrix} \frac{1}{2} & 0 & \dots & 0 & \frac{1}{4} \\ 0 & \frac{1}{2} & \dots & 0 & \frac{1}{4} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & \frac{1}{2} & \frac{1}{4} \\ \frac{1}{4} & \frac{1}{4} & \dots & \frac{1}{4} & 0 \end{bmatrix} \bullet X \\ \text{s.t.} & \operatorname{diag}(X) = e \\ & (e_{i,j,n+1} e_{i,j,n+1}^T) \bullet X = 1, \ (i,j) \in I\!\!E \\ & X \succeq 0, \end{array} \tag{7.13}$$

where the vector $e_{i,j,n+1}$ has value 1 in the position k where $k \in \{i, j, n+1\}$. Note that the non-convex constraint $rank(X) = 1$ would make this problem equivalent to (7.12).

The Benson and Ye randomized algorithm for generating stable sets [3] begins in a similar way to the Goemans-Williamson approach to find a $\{-1, 1\}$-solution $\bar{x}$. The approximate stable set will be the set of vertices with the same sign as $\bar{x}_{n+1}$. For arbitrary graphs, the constraint corresponding to the edges of the graph will be satisfied with probability of more than 91% [4]. In order to find a locally maximum stable set, the algorithm is followed by checking the edge constraint of (7.12). If $|\bar{x}_i + \bar{x}_j + \bar{x}_{n+1}| \neq 1$, for some $(i,j) \in I\!\!E$, change the sign of either $\bar{x}_i$ or $\bar{x}_j$. Note that this violation can happen only if both $\bar{x}_i$ and $\bar{x}_j$ have the same sign as $\bar{x}_{n+1}$. Obviously, solutions of (7.3) satisfy the constraints of (7.13). The implementation of the transformations (7.1) and (7.2) to the solutions of (4.6) and (7.13) leads us to the following corollary:

Corollary 1 *Let M denote the coefficient matrix in the objective function of (7.13) and let $\bar{X}_B$ be an optimal solution of (7.13), then we have $\theta'(G) \leq M \bullet \bar{X}_B = \theta(G)$.*

Proof. The inequality follows by establishing the equality. The equality is proved by Lemmas 13 and 14 below and a theorem from [20] stating that the optimal value of (7.14) is equal to $\theta(G)$. □

Lemma 13 *Let X^*, x^* be an optimal solution of*

$$\max \left\{ \ e^T x \ \middle| \begin{bmatrix} X & x \\ x^T & 1 \end{bmatrix} \succeq 0, \ x = \operatorname{diag}(X), \ X_{ij} = 0 \ \ (i,j) \in I\!\!E \right\} \tag{7.14}$$

(which is the relaxation of (4.6) and equivalent to (4.8)). Then the matrix

$$\bar{X} = \begin{bmatrix} \tilde{X} & \tilde{x} \\ \tilde{x}^T & 1 \end{bmatrix},$$

where

$$\begin{aligned} \tilde{X} &:= 4X^* + J - 2x^*e^T - 2ex^{*T}, \\ \tilde{x} &:= 2x^* - e, \end{aligned} \tag{7.15}$$

is feasible to the semidefinite relaxation (7.13) with the objective value $e^T x^*$.

Proof. A brief proof of feasibility of $\bar{X}$ will be given below: it is easy to see

$$\operatorname{diag}(\bar{X}) = e,$$

by definition of $\tilde{X}$ in (7.15). For the edge constraints of (7.13), note that for $(i,j) \in I\!\!E$ the constraint $\tilde{X}_{ij} + \tilde{x}_i + \tilde{x}_j = -1$ is equivalent to $X^*_{ij} = 0$. Therefore,

$$\begin{aligned} e_{i,j,n+1}e^T_{i,j,n+1} \bullet \bar{X} &= 2\tilde{X}_{ij} + \tilde{X}_{ii} + \tilde{X}_{jj} + 2\tilde{x}_i + 2\tilde{x}_j + 1 \\ &= 2(\tilde{X}_{ij} + \tilde{x}_i + \tilde{x}_j) + 3 = 1 \end{aligned}$$

for $(i,j) \in I\!\!E$. Theorem 1 implies positive semidefiniteness of $\bar{X}$. By the definition of $\tilde{X}$ and $\tilde{x}$ we have

$$\begin{aligned} \tilde{X} - \tilde{x}\tilde{x}^T &= 4X^* + J - 2x^*e^T - 2ex^{*T} - (4x^*x^{*T} + ee^T - 2x^*e^T - 2ex^{*T}) \\ &= 4(X^* - x^*x^{*T}) \succeq 0 \end{aligned}$$

The objective value of (7.13) with respect to $\bar{X}$ is

$$M \bullet \bar{X} = \frac{n}{2} + \frac{1}{2}(e^T\tilde{x}) = e^T x^*.$$

□

Lemma 14 *Let* $\bar{X}_B = \begin{bmatrix} X_B & x_B \\ x_B^T & 1 \end{bmatrix}$ *be an optimal solution of (7.13). The following solution*

$$\begin{aligned} x &:= \frac{x_B + e}{2}, \\ X &:= \frac{1}{4}(X_B + J + x_B e^T + e x_B^T) \end{aligned} \tag{7.16}$$

constitutes a feasible solution for (7.14) with the objective value

$$e^T x = e^T(\frac{x_B + e}{2}) = \frac{n}{2} + \frac{1}{2}(e^T x_B) = M \bullet \bar{X}_B.$$

Proof. Feasibility of the solution X, x defined in (7.16) can be shown in a similar way using the definition of X and x. □

Remark 14 *The transformation (7.15) can be applied to an optimal solution X^*,x^* of (4.6) in order to construct a feasible solution*

$$\bar{X} := \begin{bmatrix} \tilde{X} & \tilde{x} \\ \tilde{x}^T & 1 \end{bmatrix}$$

to (7.13) with the objective value $e^T x$ $(= \theta'(G))$. Let the symmetric positive semidefinite square root of $\bar{X}$ have the decomposition $\bar{X}^{1/2} = QR$ where R is an upper triangular matrix with nonnegative diagonal elements. Then

$$\bar{X} = \bar{X}^{1/2}\bar{X}^{1/2} = (\bar{X}^{1/2})^T\bar{X}^{1/2} = (QR)^T(QR) = R^TR.$$

Let $R^T := \begin{bmatrix} L & 0 \\ w^T & q \end{bmatrix}$ *be as above,*

$$\bar{X} := \begin{bmatrix} \tilde{X} & \tilde{x} \\ \tilde{x}^T & 1 \end{bmatrix} = \begin{bmatrix} L & 0 \\ w^T & q \end{bmatrix} \begin{bmatrix} L^T & w \\ 0 & q \end{bmatrix} = \bar{L}\bar{L}^T$$

will result in $\tilde{X} = LL^T$ and $w^Tw + q^2 = 1$. The last inequality shows that $\| \begin{bmatrix} w \\ q \end{bmatrix} \|_2 = 1$. Note that this factorization is not unique in general. A random $\{-1, 1\}$-vector $\hat{x}$ can be constructed by applying the GW-procedure to $\bar{X}$ as $\hat{x} = sign(\bar{L}r)$. Theorem [3a] in [4] states that $Pr((\hat{x}_i + \hat{x}_j + \hat{x}_{n+1})^2 = 1) > 0.91$ for each $(i, j) \in I\!E$ which is equivalent to

$$Pr(\hat{x}_i\hat{x}_j \text{ "satisfy the stable set property" }) > 0.912. \qquad (7.17)$$

On the other band, the GW-procedure applied to the matrix $\tilde{X}$ which is the leading block of $\bar{X}$ generates a random $\{-1, 1\}$-vector $\tilde{x}$ which forms the first n entries of $\hat{x}$ as follows,

$$\hat{x} = sign(\bar{L}r) = sign(\begin{bmatrix} L & 0 \\ w^T & q \end{bmatrix} \begin{bmatrix} r_1 \\ r_0 \end{bmatrix}) = sign(\begin{bmatrix} Lr_1 \\ w^Tr_1 + qr_0 \end{bmatrix}) = \begin{bmatrix} \tilde{x} \\ \hat{x}_{n+1} \end{bmatrix}.$$

Therefore, according to Theorem [3a] in [4], the random vector $\tilde{x}$ generated by the GW-procedure in Section 5.1 satisfies (7.17). Numerical observations reported in Section 8.3 suggest that this property holds even if the factorization of the matrices $\tilde{X}$ and $\bar{X}$ differ.

Chapter 8

Numerical Results

To gain some insight about the practical behavior of the branch and bound strategy for solving the maximum stable set problem presented in this thesis, some experiments are reported next. In this chapter we report the results of applying the algorithm to some graphs with known stability number and different density. The algorithm starts by solving the relaxation (4.9) as the root of the branch and bound tree or the zero level of the algorithm. As discussed in Section 5.1, we find a reliable upper bound on the stability number using the optimal solution of (4.9). The first level of the algorithm is to solve the relaxation (4.6) and transform its solution into a $\{0,1\}$-random vector via the GW-procedure and use a simple greedy strategy to make the $\{0,1\}$-vector feasible to the maximum stable set problem with a largest possible value of the objective function. This solution provides a lower bound on the stability number. In case there is a gap between the lower and upper bound, the next branching step includes adding more constraints to (4.6) to possibly improve the upper bound,*e.g.* the relaxation (6.3) discussed in Section 6.3. However, since the branch and bound tree will become deep for some problems we only consider one example in order to examine how the algorithm behaves in the branching step discussed in Section 8.2. We point out that these numerical results are generated using the Newton-CG augmented Lagrangian method, SDPNAL, applied to the SDP relaxation of the maximum stable set programming problem.

8.1 Evaluating the Performance of the Algorithm

We start with an example of an almost dense graph with many different maximum stable sets. Consider a complete graph K_p with p vertices. Then K_p is modified as follows: Each vertex in K_p is replaced by a complete bipartite graph $K_{p,p}$ with $2p$ vertices and each edge of K_p is replaced with $(2p)^2$ edges connecting each vertex of $K_{p,p}$ at the end of the edge with each vertex of $K_{p,p}$ at the other end of the edge. One will end up with a graph denoted by G_p with $2p^2$ vertices and $(2p)^2 \cdot \frac{p(p-1)}{2} + p \cdot p^2 = 2p^4 - p^3$ edges where the factor $\frac{p(p-1)}{2}$ is the number of edges of the complete graph K_p and the factor p^2 is of the complete bipartite graph $K_{p,p}$. Clearly, a complete bipartite graph $K_{m,n}$ has a maximum stable set of size $max\{m, n\}$. Therefore, the way in which the graph G_p is constructed leads us to a graph with $2p^2$ maximum stable sets where each one has size p.

Tables 8.1 and 8.2 list the results of applying the algorithm proposed in this thesis to find the stability number of the graph G_p for different values of p and to some well known Dimacs problems (“ DIMACS-Testset ”, [36]), respectively.

In Table 8.1, level 1 states that the algorithm returns the solution for the maximum stable set problem after one step. *time* is the time used by the algorithm to get a solution. *density* displays the density of the graph G_p.

p	density	$\alpha(G)$	level	time
50	0.9998	50	1	23889.0
30	0.9994	30	1	3607.6
20	0.9987	20	1	410.6

Table 8.1: Solution times for G_p

Figure 8.1 displays p and the associated running time as a log-log scale. For a reference, the line $time = 5.4 \cdot 10^{-4} \cdot p^{4.5}$ is plotted in Figure 8.1.

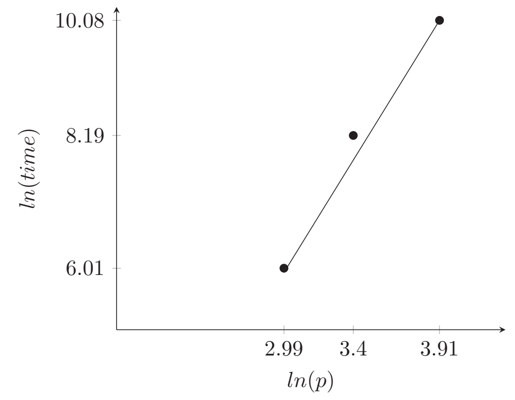


Figure 8.1: Solution times for G_p, log-log scale

The results reported in Table 8.2 below are related only to the first level of the algorithm since the branching trees for some problems in Table 8.2 are too large to allow for a full solution. The first column of Table 8.2 refers to the name of the instances. N gives the number of nodes and $|\mathbb{E}|$ gives the number of edges of the given graph. *time* is the CPU time taken to solve (4.9) and (4.6) in seconds. L_α and U_α give a lower and an upper bound on $\alpha(G)$ in level 1. *density* shows the density of the complement graph. *level* displays the number of the level at which the proposed algorithm returns an upper and lower bound to be equal. Level “ – ” means more branching steps are needed to reach a solution.

DIMACS graphs	N	$\lvert\mathbb{E}\rvert$	density	$\alpha(G)$	level	L_α	U_α	time
c-fat200-1.clq	200	18366	0.9229	12	1	12	12	26.17
c-fat200-2.clq	200	16665	0.8374	24	1	24	24	17.40
c-fat200-5.clq	200	11427	0.5742	58	–	58	60	13.94
c-fat500-1.clq	500	120291	0.9643	14	1	14	14	612.68
c-fat500-2.clq	500	115611	0.9267	26	1	26	26	338.57
c-fat500-5.clq	500	101559	0.8141	64	1	64	64	276.31
c-fat500-10.clq	500	78123	0.6262	126	1	126	126	123.74
brock200-1	200	5066	0.2546	21	–	20	27	32.07
brock200-4	200	6811	0.3423	17	–	16	21	30.06
keller4	171	5100	0.3509	11	–	11	13	31.54
keller5	776	74710	0.2485	27	–	27	30	7165.60
keller6	3361	1026582	0.1818	≥ 59	–	45	63	40669.53

Table 8.2: Dimacs Clique Benchmarks

Remark 15 *The Dimacs problems are formulated as maximum clique problems. As the approach of this thesis considers maximum stable set problems, the results reported in Table 8.2 refer to the complement graph of the Dimacs graphs.*

Remark 16 *For the graphs in Table 8.2, the application of the first three steps of the GW-procedure provides the best solution for the maximum stable set problem except for brock200-1, brock200-4 and keller6. In order to improve the lower bounds of these three graphs we apply the fourth step of case 2 on page 36. For these examples, the Step 4 modifies only the solution of keller6 by one.*

Remark 17 *SDPNAL is a software designed to solve SDPs of large dimensions, but there is one drawback to this software. This algorithm is unable to meet the preset primal/dual infeasibility conditions for some problems, even solvable problems like relaxation (4.9). In this case SEDUMI or SDPT3 are alternative software packages, however, as sparsity is not fully exploited by these programs the problem size for these algorithms is limited.*

8.2 Analysis of the Branching Step

In this section the practical behavior of different branching steps is illustrated based on the graph *brock200-1*. As seen in Table 8.2, there is a gap between lower and upper bound on $\alpha(G)$ for *brock200-1*. Thus, a branching step is incorporated into the algorithm in order to get a more accurate approximate solution to the maximum stable set problem. First, a vector v is determined via (6.5), (6.7) or (6.8). Then, for each value of γ, the problem (6.3) will be solved in the branching step. The following Tables 8.3, 8.4 and 8.5 compare the results when we solve

(6.3) for the graph *brock200-1* with the different versions of the vector v generated from the solutions of (6.5), (6.7) and (6.8).

Table 8.3 reports the optimal value of the relaxation (6.3) obtained with SDPNAL for different values of $\gamma = -13, \ldots, 10$ where the vector v is determined by (6.8). The duality gap for these examples was in the range of $4.18e^{-5}$ to $6.31e^{-7}$. For all subproblems the dual infeasibility of the solution generated by SDPNAL was in the range $1.5e^{-7}$ to $7.7e^{-7}$ which is about the standard stopping tolerance of SDPNAL. If these solutions are interpreted as nearly dual feasible, the results of SDPNAL imply that all values of γ can be deleted from the branch-and-bound tree when trying to reduce the upper bound from 26 to 25, except from the values $\gamma = 1, 0, -1, -2, -3$ for which the upper bound is $\alpha(G) \leq 26$. For these values, a further branching is necessary. However, as the stability number is 21, all the cases $\gamma = -10, -9, \ldots, 6$ would need further branching if the optimal value is to be found with this branching strategy. This possibility is discussed further following relaxation (8.1) below.

γ	opt.val	γ	opt.val	γ	opt.val	γ	opt.val
-13	18.99	-7	23.67	-1	26.68	5	23.13
-12	20.00	-6	24.38	0	26.61	6	21.89
-11	20.80	-5	25.09	1	26.34	7	20.50
-10	21.60	-4	25.71	2	25.88	8	19.11
-9	22.31	-3	26.20	3	25.18	9	18.00
-8	22.98	-2	26.54	4	24.24	10	0

Table 8.3: v is the solution of (6.8)

The results of solving the relaxation (6.3) where v is the solution of (6.5) and (6.7) are reported in Tables 8.4 and 8.5. The solvability of (6.3) in both cases is similar to the one reported in Table 8.3.

γ	opt.val	γ	opt.val	γ	opt.val
-13	18.99	-7	23.93	-1	26.60
-12	19.99	-6	24.64	0	26.22
-11	20.83	-5	25.41	1	25.43
-10	21.66	-4	26.03	2	24.10
-9	22.43	-3	26.47	3	22.45
-8	23.18	-2	26.68	4	20.33

Table 8.4: v is the solution of (6.5)

Our numerical experiments suggest no significant difference, however, less branching nodes are needed in the next step when we apply the second version of v generated from the solution of (6.7) discussed in case 2 of Section 6.3 instead of the other two versions of v.

γ	opt.val	γ	opt.val	γ	opt.val
-8	0	-2	25.73	4	23.54
-7	0	-1	26.48	5	21.91
-6	0	0	26.81	6	20.12
-5	19.90	1	26.70	7	18.11
-4	22.71	2	26.07	8	0
-3	24.99	3	24.97	9	0

Table 8.5: v is the solution of (6.7)

As discussed, further branching is needed for *brock200-1* in order to reduce the gap between the upper and lower bound on the stability number. To this end, the data from one of the Tables 8.3, 8.4 and 8.5 can be used. Consider Table 8.4 for instance. The value $\gamma = -2$ for which the relaxation (6.3) attains its maximum value 26.68 is selected as the starting point. One of the methods which can be applied for further branching is to add the additional constraint $Xv' = \gamma' x$ to the relaxation (6.3) where v', γ' are chosen the same way as v and γ. The new relaxation is defined as

$$\max \{ \ e^T x \mid \ \hat{X} \succeq 0, \ \hat{X} \geq 0, \ A_G \bullet X = 0, \ x = \operatorname{diag}(X), \ Xv = \gamma x, \ Xv' = \gamma' x \}. \qquad (8.1)$$

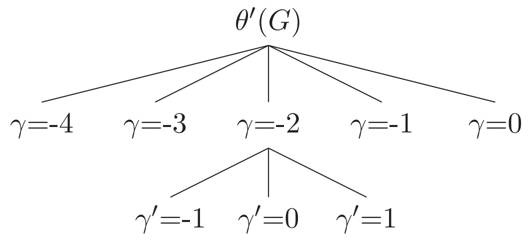
We point out that v and γ are known variables from the last step.

γ'	opt.val	γ'	opt.val	γ'	opt.val
-10	17.23	-3	25.01	4	23.76
-9	19.00	-2	25.66	5	22.63
-8	20.31	-1	26.08	6	21.38
-7	21.41	0	26.23	7	20.10
-6	22.42	1	26.06	8	19.00
-5	23.33	2	25.56	9	0
-4	24.20	3	24.75		

Table 8.6: v' is the solution of (6.8)

Table 8.6 reports the optimal value of the relaxation (8.1) for $\gamma = -2$ and its associated vector v. The same process would need to be carried out for $\gamma = -4, -3, -1, 0$. In Table 8.6, the branching would need to be continued in the same way until the lower and upper bound on $\alpha(G)$ coincide. The following diagram of the example above provides an overview of the structure of the algorithm, just for the aim to reduce the upper bound for $\alpha(G)$ to 25.

Only two branching steps are considered in the example above. It can be inferred from these experiments that the branch and bound tree will expand in depth and for each node may have many children to deal with. Consequently, the running time may be exponential – which



is not surprising given that the problem is NP-complete. The times used by SDPNAL on a standard computer for generating the results of Tables 8.3, 8.4, and 8.5 are 2523.03, 1601.4, and 1286.7 in seconds respectively.

The other method we can apply for the branching step instead of solving relaxation (8.1) is the one discussed in Section 6.1. We start the algorithm by setting $\beta = 26$ and generating a vector $\tilde{x} \in \{0,1\}^n$ with $e^T\tilde{x} \geq \beta$. Here, $\tilde{x}$ can be constructed in the ways discussed in Section 6.1.1. As seen in Table 8.2, the best maximum stable set found for graph *brock200-1* is of size 20. In order to apply the first choice of $\tilde{x}$ we start with a maximum stable set of size 20 as the set S and find a clique covering of S. There are 98 relaxations to be solved in this case. Among these 98 relaxations excluding node j with $\tilde{x}_j = 0$, the best and the worst solutions are 24.38 and 21.93. This observation has resulted in improving the upper bound from 27 to 25. On the other hand, the best maximum stable set given by the GW-approach is of size 21. Therefore, we also improve the lower bound of *brock200-1* from 20 to 21 (which happens to be the optimal value).

There are two other methods of choosing $\tilde{x}$ in Section 6.1.1. For using the methods 2 and 3 we set $e^T\tilde{x} = \beta$. There are 174 subgraphs and therefore 174 subproblems to be solved in order to get a possibly improved upper bound U_α on $\alpha(G)$. We observed that the relaxation (4.6) applied to the subgraphs G_i, $i = 1, \ldots, 174$ provides an upper bound with the maximum value of 24. Assuming that there are no significant rounding errors this shows that the maximum stable set of G_i with node i has size at most 25. Therefore, the upper bound on $\alpha(G)$ can be reduced by 1. We note that the nodes i with $\tilde{x}_i = 1$ have already been investigated in Step 1 in Section 6.1. In contrast, as observed from Table 8.6 the relaxation (8.1) does not improve the upper bound on the stability number of *brock200-1*.

Of the two methods used for the branching step, the relaxation (8.1) and the one discussed in Section 6.1, the latter seems to return a solution to the maximum stable set problem in a lesser number of levels than the former. The downside is that the computation time for solving 174 problems of the form (4.6) is about twice the computation time of solving 20 problems of the form (8.1). On the other hand, the computation time for solving 98 relaxations of the form (4.6) is approximately the same as the computation time of 20 problems of the form (8.1).

Remark 18 *Note that the relaxation (6.3) is an equivalent mathematical formulation of the method discussed in Step 2 in Section 6.1 when we set $v = e_i$ and $\gamma = 0$ in (6.3).*

8.3 The Probability of Violated Edges in the $\{0,1\}$-Solution Made by Goemans and Williamson

As shown in [11], the probability that a $\{-1,1\}$-solution generated by the rounding procedure proposed by Goemans and Williamson for the max-cut problem with nonnegative weights achieves an objective value of at at least 87% of the optimal value. The same is true if the solution is represented by a 0-1 variable. In the context of the maximum stable set problem not only the objective value but also the feasibility of a generated $\{0,1\}$-solution will be of interest. In particular it is interesting to understand the probability that a given edge in $\mathbb{E}$ is contained in the approximate stable set generated by rounding.
In our examples, the case where there is no violated edge typically is when the set generated by the GW-approach has rather small cardinality far from the maximum stable set. We assume that the GW rounding procedure generates a set of size k and let $|E_k|$ denote the number of violated edges of the set. As we discussed before, our approach towards finding a locally maximum stable set is similar to the approach proposed by Benson and Ye. Therefore, the quotient $P_k := \frac{|E_k|}{|\mathbb{E}|}$ should be less than or equal 8.8% in the average by Theorem 3a in [4].
The column n of the table displays the cardinality of the vertex set in a given graph. For each dimension n, a total of $t_n = 10$ different random graphs with fixed edge density were generated, and for each random graph, a total of $p = 100$ different random Goemans-Williamson vectors were generated. Therefore, for each dimension there are 10×100 random $\{0,1\}$-vectors generated by the GW-test.

	edge density							
	20%				80%			
n	min	max	mean	emp	min	max	mean	emp
10	981	0.111	0.002	0	410	0.205	0.045	59
50	204	0.197	0.051	0	580	0.241	0.008	28
100	40	0.215	0.069	0	534	0.233	0.009	60
200	44	0.223	0.066	8	198	0.241	0.042	151
500	53	0.229	0.061	15	214	0.245	0.039	212

Table 8.7: Estimate the probability

Column *min* displays the number of examples at which the GW-approach provides a feasible solution to the maximum stable set problem. Columns *max* and *mean* give the maximum number of violated edges of the $\{0,1\}$-vectors generated by the Goemans-Williamson rounding procedure and the average of P_k's, respectively. There is a possibility that the GW-approach generates an empty set (equivalently, a set of size n). *emp* shows the incidences of empty sets. We are not interested in the number of feasible or empty sets which are irrelevant to our discussion, whereas they can be used as performance indicators for the GW-procedure. The last four entries of the last row of Table 8.7 are to be read as follows: For n=500 and 80% edge density, 21.2% of the stable sets generated by the Goemans-Williamson approach were empty, another 21.4% were (rather small) stable sets and in the average, the sets generated contained about 3.9% edges.

Appendix A

It can be proved by Weyl's theorem that the symmetric eigenvalue problem has condition number 1. In this thesis we assumed that the eigenvalue decomposition is carried out with relative accuracy $\leq 100 \cdot \epsilon$, where ϵ is the machine accuracy. This value is obtained by comparing the eigenvalues of a matrix given by matlab to the actual eigenvalues of that matrix. In what follows we show how to obtain an estimate for this accuracy: Let us assume that $U \in \mathbf{R}^{n \times n}$ be a random orthogonal matrix and $D \in \mathbf{R}^{n \times n}$ be a random diagonal matrix. By the eigenvalue decomposition of the matrix we have $A = UDU^T$. We use the matlab command $d' := eig(A)$ to get the eigenvalues of the matrix A which are an approximation of the diagonal entries of D, denoted by $d := diag(D)$. The vector d has the actual eigenvalues of A and d' is computed by matlab, therefore the relative error definition gives

$$relative\ error = \frac{\|d - d'\|}{\|d\|}$$

One can easily prove that $\|d\| = \|A\|_F$. Thus, the difference between the actual value of the eigenvalues of A and their approximate value is given by

$$\|d - d'\| = (\ relative\ error) \cdot \|A\|_F.$$

The tables below report the minimum and maximum value of relative errors over a number of repetitions. For each dimension n, a total of 100 different random matrices were constructed. The table at the left shows the results for random matrices with different eigenvalues. The tables at the center and at the right are related to the instances of random matrices which have multiple eigenvalues and near multiple eigenvalues, respectively.

Remark 19 *Even if A is computed by a formula that generates symmetric matrices, e.g., $A = BB^T$; it may happen due to rounding errors, that numerically, A is not symmetric. Therefore A will be replaced by $\frac{1}{2}(A + A^T)$. If A is numerically symmetric, this update will not induce any change of A. If A is (slightly) unsymmetric, then the updated value will be symmetric and $eig(A)$ will use the symmetric eigenvalue algorithm which are well-conditioned, while the algorithms for the unsymmetric eigenvalue problem tend to be very sensitive to rounding errors.*

n	rel.error	
	min	max
5	8.9e-17	1.8e-15
10	3.5e-16	2.2e-15
50	1.5e-15	4.4e-15
100	1.9e-15	4.1e-15
500	3.1e-15	5.7e-15
1000	3.9e-15	6.4e-15

n	rel.error	
	min	max
50	5.8e-16	1.7e-15
100	8.9e-16	2.1e-15
500	2.3e-15	3.3e-15

n	rel.error	
	min	max
50	4.2e-16	1.6e-15
100	9.3e-16	1.9e-15
500	2.2e-15	3.4e-15

Table A.1: Relative error for random matrices

The following lemma shows that the eigenvalue of a symmetric matrix are not sensitive to perturbation of the matrix. In linear algebra, the condition number for the linear transformation $\phi : \mathbb{D} \to \mathbb{R}^m$ for an open set $D \subset \mathbb{R}^n$ is a number $c > 0$ such that

$$\frac{\|\phi(\tilde{x}) - \phi(x)\|}{\|\phi(x)\|} \leq c\frac{\|\tilde{x} - x\|}{\|x\|} \quad for\ all\ x, \tilde{x} \in D,$$

in conjunction with suitable norm $\| \, . \, \|$.

Lemma 15 *The condition number of symmetric eigenvalue problem is less than or equal to one.*

Proof. Let $\overrightarrow{\lambda}$ be the vector of eigenvalues of a symmetric matrix $A \in \mathbb{R}^{n\times n}$. The generalized definition of condition number in [28] implies that

$$\frac{\|\overrightarrow{\lambda}(A+E) - \overrightarrow{\lambda}(A)\|_2}{\|\overrightarrow{\lambda}(A)\|_2} \leq c\frac{\|E\|_F}{\|A\|_F},$$

where E denotes a perturbation of A and c is the condition number. Note that

$$\|A\|_F = \|U\Lambda U^T\|_F = \|\Lambda\|_F = \|\overrightarrow{\lambda}(A)\|_2.$$

On the other hand, from [[27], Corollary 4.13] we have

$$\|\overrightarrow{\lambda}(A+E) - \overrightarrow{\lambda}(A)\|_2 \leq \|E\|_F.$$

□

Bibliography

[1] A. Schrijver. *Combinatorial Optimization, Polyhedra and Efficiency*. Springer, 2003.

[2] F. Alizadeh, J. A. Haeberly, M. L. Overton. *Complementarity and Nondegeneracy in Semidefinite Programming*. Mathematical Programming, vol. 77, pp. 111-128, 1997.

[3] S. J. Benson, Y. Ye. *Approximating Maximum Stable Set and Minimum Graph Coloring Problems with the Positive Semidefinite Relaxation*. In: Complementarity: Applications, Algorithms and Extensions, M. Ferris, O. L. Mangasarian, J. Pang (editors), Springer, pp. 1–17, 2001.

[4] D. Bertsimas, Y. Ye. *Semidefinite Relaxations, Multivariate Normal Distributions, and Order Statistics*. In: Handbook of Combinatorial Optimization, D. Z. Du, P. M. Pardalos (editors), Kluwer Academic Publishers, Boston, pp. 1473–1491, 1998.

[5] S. Burer, R. D. C. Monteiro, Y. Zhang. *Maximum Stable Set Formulations and Heuristics Based on Continuous Optimization*. Technical Report TR00-34, Department of Computational and Applied Mathematics, Rice University, 2000.

[6] S. Butenko. *Maximum independent set problem and related problems, with applications*. A dissertation presented to the graduate school of the University of Florida in partial fulfillment of the requirements for the degree of Doctor of Philosophy, 2003.

[7] I. Dukanovic, F. Rendl. *Semidefinite programming relaxation for graph coloring and maximal clique problem*. Mathematical programming, vol. 109, pp. 345–365, 2007.

[8] H. Federer. *Geometric Measure Theory*. Springer, 1996.

[9] R. W. Freund, F. Jarre. *A Sensitivity Result for Semidefinite Programs*. Operations Research Letters, vol. 32, pp. 126-132, 2004.

[10] M. R. Garey, D. S. Johnson. *Computers And Intractability, A Guide to the Theory of NP-Completeness*. W. H. Freeman and Company (New York), 1979.

[11] M. X. Goemans, D. P. Williamson. *Improved Approximation Algorithms for Maximum Cut and Satisfiability Problems Using Semidefinite Programming*. Journal of the ACM, vol. 42, pp. 1115–1145, 1995.

[12] M. Grötschel, L. Lovász, A. Schrijver. *Complexity, Oracles, and Numerical Computation.* In: Geometric Algorithms and Combinatorial Optimization, Springer, 1988.

[13] N. Gvozdenović, M. Laurent, F. Vallentin. *Block-Diagonal semidefinite programming hierarchies for 0/1 programming.* Operations Research Letters, vol. 37, pp. 27–31, 2009

[14] J. B. Hiriart-Urruty, C. Lemaréchal. *Convex Analysis and Minimization Algorithm I: Fundamentals.* Springer, 1993.

[15] F. Jarre, J. Stoer. *Mathematische Optimierung.* Springer, 2010.

[16] E. de Klerk. *Aspects of Semidefinite Programming: Interior Point Algorithms and Selected Applications.* Kluwer Academic Publishers, 2002.

[17] J. B. Lasserre. *An explicit exact SDP relaxation for nonlinear 0-1 programs.* K. Aardal, B. Gerards (editors): International conference on Integer Programming and Combinatorial Optimization, Lecture Notes in Computer Science 2081, pp. 293–303, 2001.

[18] M. Laurent. *A comparison of the Sherali-Adams, Lovász-Schrijver and Lasserre relaxations for 0-1 programming.* Mathematics of Operations Research, vol. 28, pp. 470–496, 2003.

[19] M. Laurent. *Semidefinite Programming in Combinatorial and Polynomial Optimization.* NAW 5/9 nr. 4, 2008.

[20] M. Laurent, F. Rendl. *Semidefinite Programming and Integer Programming.* In: Handbook on Discrete Optimization, K. Aardal, G. Nemhauser, R. Weismantel (editors), pp. 393–514, Elsevier, 2005.

[21] L. Lovász. *On the Shannon capacity of a graph.* IEEE Transactions on Information Theory, IT-25:1–7, 1979.

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

[23] A. S. Nemirovsky, M. J. Todd. *Interior Point Methods for Optimization.* Acta Numerica, pp. 191–234, 2008.

[24] Y. Nesterov, A. Nemirovsky. *Interior Point Polynomial Methods in Convex Programming.* SIAM, 1994.

[25] R. T. Rockafellar. *Augmented Lagrangians and Applications of the Proximal Point Algorithm in Convex Programming.* Mathematics of Operations Research, pp. 97-116, 1976.

[26] H. D. Sherali, W. P. Adams. *A hierarchy of relaxations between the continuous and convex hull representations for zero-one programming problems.* SIAM J. Discrete Mathematics, vol. 3, pp. 411–430, 1990.

[27] G. W. Stewart, J. Sun. *Matrix Perturbation Theory.* Academic Press, Boston, 1990.

[28] J. Stoer, R. Bulirsch. *Introduction to Numerical Analysis.* Springer, 2002.

[29] J. F. Sturm. *Using SEDUMI 1.02, A MATLAB Toolbox for Optimization over Symmetric Cones.* Department of Econometrics, Tillburg University, Tillburg, The Netherlands, 1998-2001.

[30] D. Sun, J. Sun. *Semismooth Matrix-Valued Functions.* Mathematics of Operations Research, vol. 22, pp. 150–169, 2002.

[31] K. C. Toh, M. J. Todd, R. H. Tutuncu. *SDPT3-a MATLAB software package for semidefinite programming.* Optimization Methods and Software, vol. 11, pp. 545–581, 1999.

[32] M. Ulbrich. *Nonsmooth Newton-like Methods for Variational Inequalities and Constrained Optimization Problems in Function Spaces.* Habilitation, Technische Universität München, 2002.

[33] H. Wei, H. Wolkwicz. *Generating and measuring instances of hard semidefinite programs.* Mathematical Programing, vol. 125, pp. 31–45, 2010.

[34] E. A. Yildirim, X. Fan-Orzechowski. *On Extracting Maximum Stable Set in Perfect Graphs Using Lovász's Theta Function.* Computatioal Optimization and Application, vol. 33, pp. 229-247, 2006.

[35] X. Zhao, D. Sun, K. Toh. *A Newton-CG Augmented Lagrangian Method for Semidefinite Programming.* SIAM Journal Optimization, vol. 20, pp. 1737–1765, 2010

[36] " DIMACS-Testset." Retrieved from `http://dimacs.rutgers.edu/pub/challenge/graph/benchmarks/clique/`.

Statment of Originality

Die hier vorgelegte Disseration habe ich eigenständig und ohne unerlaubte Hilfe angefertigt. Die Disseration wurde von der vorgelegten oder in ähnlicher Form noch bei keiner anderen Institution eingereicht. Ich habe bisher keine erfolglosen Promotionsversuche unternommen.

I do herewith declare that the material contained in this dissertation is an original work performed by me without illegitimate help. The material in this thesis has not been previously submitted for a degree in any University.

Düsseldorf, February 2018

Fatemeh Baniasadirad