

## Semidefinite programming approaches for stable set and max-cut problems

Authors	Sinjorgo, Lennart
Published in	CentER Dissertation Series
DOI	<a href="https://doi.org/10.26116/tisem.63859127">10.26116/tisem.63859127</a>
Publication Date	2026
Document Version	publishersversion
Link	<a href="https://research.tilburguniversity.edu/en/publications/b5eb84a2-4033-469e-8288-00285780ac88">https://research.tilburguniversity.edu/en/publications/b5eb84a2-4033-469e-8288-00285780ac88</a>
Citation	Sinjorgo, L 2026, 'Semidefinite programming approaches for stable set and max-cut problems', Doctor of Philosophy, Tilburg University, Tilburg. <a href="https://doi.org/10.26116/tisem.63859127">https://doi.org/10.26116/tisem.63859127</a>
Download Date	2026-05-11 02:30:38
Rights	<p>General rights</p> <p>Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.</p> <ul style="list-style-type: none"> <li>- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.</li> <li>- You may not further distribute the material or use it for any profit-making activity or commercial gain</li> <li>- You may freely distribute the URL identifying the publication in the public portal"</li> </ul> <p>Take down policy</p> <p>If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.</p>



# Semidefinite programming approaches for stable set and max-cut problems

LENNART SINJORGÓ



# Semidefinite programming approaches for stable set and max-cut problems

Proefschrift ter verkrijging van de graad van doctor aan Tilburg University op gezag van de rector magnificus, prof. dr. W.B.H.J. van de Donk, in het openbaar te verdedigen ten overstaan van een door het college voor promoties aangewezen commissie in de aula van de Universiteit op

vrijdag 27 februari 2026 om 13.30 uur

door

Lennart Matteo Sinjorgo,  
geboren te 's-Hertogenbosch

Promotores:            prof. dr. ir. R. Sotirov (Tilburg University)  
                              prof. dr. J.C. Vera Lizcano (Tilburg University)

Promotiecommissie:   dr. C. Hojny (Eindhoven University of Technology)  
                              prof. dr. F. Jarre (Heinrich-Heine-Universität Düsseldorf)  
                              prof. dr. M. Laurent (Tilburg University)  
                              dr. mr. S.C. Polak (Tilburg University)  
                              prof. dr. F. Vallentin (University of Cologne)

De Bondt Grafimedia

©2026 Lennart Matteo Sinjorgo, The Netherlands. All rights reserved. No parts of this thesis may be reproduced, stored in a retrieval system or transmitted in any form or by any means without permission of the author. Alle rechten voorbehouden. Niets uit deze uitgave mag worden vermenigvuldigd, in enige vorm of op enige wijze, zonder voorafgaande schriftelijke toestemming van de auteur.

# Acknowledgements

During my time as a PhD candidate, I have been lucky to be supported and guided by many people.

I would like to thank my supervisor Renata Sotirov, for, among other things, her never-ending support, attention to detail, and her many fruitful suggested research directions. Thank you for considering my, admittedly, often typo-riddled drafts in great detail, and thank you for suggesting countless improvements along the way.

I would like to thank my co-supervisor Juan Vera for his valuable feedback. Your suggestions regarding numerical results, and the presentation thereof, have been hugely helpful. This thesis has also benefited from many improvements that have been suggested to me by my PhD committee. I would like to thank the committee for their time and careful reading.

I would also like to thank my co-authors, for providing valuable feedback and fresh perspectives. In particular, I would like to thank Miguel Anjos, for hosting me at the beautiful University of Edinburgh.

I would like to thank my K420 office mates over the years. Thank you for the many jokes, for brightening up the office, and for creating a friendly atmosphere that I will surely miss. I would also like to thank the other Tilburg University PhD candidates, for making the many work parties more enjoyable, for collaborating in teaching, and for joining smash-fridays.

I would like to thank my parents, and my sister, who have always motivated and supported me since the beginning. I would also like to thank my partner, for always being there for me. Thank you for always listening.

# Contents

<b>Notation and abbreviations</b>	<b>vii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 An SDP problem and its dual . . . . .	1
1.2 Polynomial optimization . . . . .	2
1.3 Applications of SDP to selected problems . . . . .	6
1.4 The ADMM and its variants . . . . .	13
1.5 Overview of the thesis . . . . .	15
1.6 Contributions to the literature . . . . .	17
<b>2 The generalized <math>\vartheta</math>-number and related problems for highly symmetric graphs</b>	<b>19</b>
2.1 Preliminaries . . . . .	21
2.2 $\vartheta_k(G)$ and $\chi_k(G)$ formulations and their relation . . . . .	23
2.3 The sequence $(\vartheta_k(G))_{k \in [n]}$ . . . . .	25
2.4 Graph products and the generalized $\vartheta$ -number . . . . .	29
2.5 Value of $\vartheta_k$ for some graphs . . . . .	32
2.6 Strongly regular graphs . . . . .	42
2.7 Orthogonality graphs . . . . .	47
2.8 New bounds on $\chi_k(G)$ . . . . .	49
2.9 Conclusions . . . . .	54
<b>3 Cuts and semidefinite liftings for the complex cut polytope</b>	<b>56</b>
3.1 Preliminaries . . . . .	57
3.2 Framework for finding valid inequalities for $\text{CUT}_m^n$ . . . . .	60
3.3 An exact description of $\text{CUT}_3^3$ . . . . .	68
3.4 Efficiently reformulating a class of CSDPs . . . . .	71
3.5 Second semidefinite lifting of $\text{CUT}_\infty^n$ . . . . .	74
3.6 Extreme points of $\mathcal{E}_m^3$ . . . . .	85
3.7 Numerical results . . . . .	88
3.8 Conclusions . . . . .	92

<b>4</b>	<b>Improved approximation ratios for the quantum max-cut problem on general, triangle-free and bipartite graphs</b>	<b>95</b>
4.1	Preliminaries . . . . .	98
4.2	SDP bounds on $\lambda_{\max}(H_G)$ . . . . .	99
4.3	Improved analysis of a QMC approximation algorithm . . . . .	101
4.4	New approximation algorithm on triangle-free graphs . . . . .	108
4.5	New approx. algorithm on bipartite graphs . . . . .	117
4.6	Conclusions . . . . .	123
<b>5</b>	<b>SDP bounds on the stability number via ADMM and intermediate levels of the Lasserre hierarchy</b>	<b>125</b>
5.1	Preliminaries . . . . .	127
5.2	The Lasserre hierarchy for the stable set problem . . . . .	127
5.3	The ADMM for computing $\alpha^{\mathcal{B}}(G)$ . . . . .	129
5.4	A dynamic basis selection method and ADMM initialization . . . . .	133
5.5	Numerical results . . . . .	135
5.6	Conclusions . . . . .	146
<b>6</b>	<b>On solving the MAX-SAT problem using sum of squares</b>	<b>147</b>
6.1	Preliminaries . . . . .	149
6.2	MAX-SAT formulations and relaxations . . . . .	151
6.3	The SAT problem as a semidefinite feasibility problem . . . . .	152
6.4	Sum of squares and the MAX-SAT problem . . . . .	155
6.5	Resolution and monomial bases . . . . .	160
6.6	Relating sum of squares and method of moments . . . . .	162
6.7	The PRSM for the MAX-SAT problem . . . . .	165
6.8	The weighted partial MAX-SAT problem . . . . .	169
6.9	SOS-MS: Algorithm description . . . . .	173
6.10	Numerical results . . . . .	178
6.11	Conclusions . . . . .	182
<b>A</b>	<b>Supplementary data</b>	<b>191</b>
A.1	Scaled form of the ADMM . . . . .	191
A.2	Runtimes per MAX-3-SAT instance . . . . .	191
A.3	Branching process for the partial MAX-2-SAT problem . . . . .	193
A.4	Search tree for the partial MAX-2-SAT problem . . . . .	195
A.5	Reflection symmetries of $\mathcal{U}_m$ and $\text{CUT}_m^n$ . . . . .	200
A.6	Facet enumeration of $\mathcal{V}(\text{CUT}_3^3)$ . . . . .	201
A.7	Computational details of proof of Lemma 4.17 . . . . .	201
<b>B</b>	<b>Technical lemmas and proofs</b>	<b>204</b>
B.1	Proofs from Chapter 2 . . . . .	204
B.2	Proofs from Chapter 3 . . . . .	204
B.3	Proofs from Chapter 4 . . . . .	211
	<b>Academic summary</b>	<b>218</b>

<b>Academische samenvatting</b>	<b>220</b>
<b>Bibliography</b>	<b>222</b>

# Notation and abbreviations

## Sets

$\mathcal{U}_n$	set of the complex $n$ th roots of unity
$\mathbb{C}$	set of complex numbers
$\text{Conv}(S)$	convex hull of elements in $S$ (also written as $\text{Conv } S$ )
$\mathcal{H}^n$	set of $n \times n$ complex Hermitian matrices
$\mathcal{H}_+^n$	set of $n \times n$ complex Hermitian PSD matrices
$\mathbb{N}_d^n$	$:= \left\{ \alpha \in (\mathbb{N} \cup \{0\})^n : \sum_{i \in [n]} \alpha_i \leq d \right\}$
$[n]$	set of integers $\{1, \dots, n\}$
$\binom{[n]}{\leq k}$	$:= \{\beta \subseteq [n] :  \beta  \leq k\}$ for integers $n$ and $k$
$\mathbb{R}$	set of real numbers
$\mathbb{R}_+$	$:= \{x \in \mathbb{R} : x \geq 0\}$
$\mathcal{S}^n$	set of $n \times n$ real symmetric matrices
$\mathcal{S}_+^n$	set of $n \times n$ real symmetric positive semidefinite matrices

**Linear algebra**

$\mathbf{0}_n$ or $\mathbf{0}$	all-zeroes column vector of length $n$ , or of fitting size
$\mathbf{1}_n$ or $\mathbf{1}$	all-ones column vector of length $n$ , or of fitting size
$\det(\cdot)$	determinant of a square matrix
$\text{Diag}(\cdot)$	$\text{Diag}(a)$ is the diagonal matrix with $a$ as its diagonal
$\text{diag}(\cdot)$	adjoint operator of $\text{Diag}(\cdot)$
$(\cdot)^{\text{H}}$	Hermitian/conjugate transpose
$\mathbf{I}_n$ or $\mathbf{I}$	$n \times n$ identity matrix or identity matrix of fitting size
$\mathbf{J}_n$ or $\mathbf{J}$	$n \times n$ matrix of ones or matrix of ones of fitting size
$\Lambda(\cdot)$	eigenspectrum of a square matrix
$M \succeq \mathbf{0}$	matrix $M$ is positive semidefinite
$M \succeq N$	matrix $M - N$ is positive semidefinite
$\langle M, N \rangle$	trace inner product $\text{tr}(M^{\text{H}}N)$
$\ M\ $	Frobenius norm of matrix $M$ , i.e., $\sqrt{\text{tr}(M^{\text{H}}M)}$
$M \odot N$	Hadamard product of matrices $M$ and $N$
$M \otimes N$	Kronecker product of matrices $M$ and $N$
$\mathcal{P}_{\mathcal{F}}(\cdot)$	projection onto set $\mathcal{F}$ , i.e., $\mathcal{P}_{\mathcal{F}}(X) := \arg \min_{Y \in \mathcal{F}} \ X - Y\ $
$\text{rk}(\cdot)$	rank of input matrix
$(\cdot)^{\text{T}}$	transpose
$\text{tr}(\cdot)$	the trace of a square matrix
$\text{vech}(X)$	vector of all $n(n+1)/2$ lower triangular entries of $X \in \mathcal{S}^n$

**Graph theory**

$A_G$	$\{0, 1\}$ adjacency matrix of graph $G$
$\alpha(G)$	independence/stability number of graph $G$
$C_n$	cycle graph on $n$ vertices
$\text{deg}(i)$	degree of vertex $i$ , i.e., the number of vertices adjacent to $i$
$E(G)$	edge set of graph $G$
$G = (V, E)$	simple undirected graph with vertex set $V$ and edge set $E$
$\mathbb{G}_n$	set of all simple, undirected, unweighted graphs on $n$ vertices
$\overline{G}$	complement graph of $G$
$K_n$	complete graph on $n$ vertices
$L_G$	Laplacian matrix of $G$
$N(i)$	set of vertices adjacent to vertex $i$
$[n]_G$	$:= \{\beta \subseteq [n] : \beta \text{ is stable in } G\}$ , where $V(G) = [n]$
$\omega(G)$	clique number of graph $G$
$\text{srg}(n, d, \lambda, \mu)$	strongly regular graph
$\tau(G)$	vertex cover number of graph $G$
$V(G)$	vertex set of graph $G$
$\chi(G)$	chromatic number of graph $G$

**Polynomials**

$\deg(\cdot)$	degree of polynomial
$\mathbb{F}[x]$	polynomials with coeffs. in the field $\mathbb{F}$ , in commutative variables
$\mathbb{F}[x]_d$	polynomials in $\mathbb{F}[x]$ of degree at most $d$ , $d \in \mathbb{N}$ .
$\mathbb{F}\langle x \rangle, \mathbb{F}\langle x \rangle_d$	noncommutative variants of $\mathbb{F}[x]$ and $\mathbb{F}[x]_d$
$\langle f_i, \dots, f_m \rangle_k$	$:= \{ \sum_{i=1}^m g_i f_i h_i : g_i, h_i \in \mathbb{F}\langle x \rangle, \deg(g_i f_i h_i) \leq k \quad \forall i \in [m] \}$
$s(d)$	$:= \binom{n+d}{n}$ , size of $\mathbf{v}_d(x)$ , where $x$ consists of $n$ variables
$\mathbf{v}_d(x)$	vector of monomials that form a basis of $\mathbb{F}[x]_d$ , $d \in \mathbb{N}$
$\mathbf{x}^{\mathcal{B}}$	vector of monomials $\left( \prod_{i \in \beta} x_i \right)_{\beta \in \mathcal{B}}$ , for $\mathcal{B} \subseteq \binom{[n]}{\leq k}$

**Miscellaneous**

$\mathbb{1}_\beta$	for some $\beta \subseteq [n]$ , $\mathbb{1}_\beta \in \{0, 1\}^n$ , with $(\mathbb{1}_\beta)_i = 1$ iff. $i \in \beta$
$\text{Arg}(\cdot)$	angle in $[0, 2\pi)$ of a complex number, with $\text{Arg}(0) = 0$
$\mathbb{E}[\cdot]$	expectation operator
$\exp(\cdot)$	exponential function
$\mathbb{I}(\cdot)$	for $\beta \subseteq [n]$ , $\mathbb{I}(\beta) = 1$ if $ \beta  = 1$ , and 0 otherwise
$\text{Im}(\cdot)$	imaginary part of a complex matrix
$\mathbf{i}$	imaginary unit
$\text{Re}(\cdot)$	real part of a complex matrix
$\text{sgn}(\cdot)$	sign of a real number (with $\text{sgn}(0) = 1$ )
$\vee$	logical disjunction
$\wedge$	logical conjunction
$\triangle$	symmetric difference, i.e., $A \triangle B := (A \cup B) \setminus (A \cap B)$
$ \cdot $	cardinality of a set, or absolute value: $ z  = \sqrt{z^H z}$
$\lfloor \cdot \rfloor$	floor function
$\lceil \cdot \rceil$	ceiling function
$\text{round}(\cdot)$	round to nearest integer function

**Abbreviations**

ADMM	alternating direction method of multipliers
CNF	conjunctive normal form
CSDP	complex semidefinite program(ming)
EGF	extremal Gram factor
EPR	Einstein, Podolsky, Rosen
IPM	interior-point method
LP	linear program(ming)
MAX-SAT	maximum satisfiability
MIMO	multiple-input multiple-output
MLE	maximum likelihood estimator
POP	polynomial optimization problem
PRSM	Peaceman-Rachford splitting method
PSD	positive semidefinite
QMC	quantum max-cut
SAT	satisfiability
SDP	semidefinite program(ming)
SOS	sum of squares
s.t.	such that, or, subject to
UGC	unique games conjecture
w.l.o.g.	without loss of generality

# 1 Introduction

Mathematical optimization is the study of solution methods for optimization problems. An optimization problem is to optimize (i.e., minimize or maximize) a given real-valued objective function, over a given feasible set. Such optimization problems arise in many fields of science, such as operations research, finance, engineering, and machine learning. Optimization problems can be divided in classes. For example, when the feasible set is finite, we speak of discrete optimization. When the objective function is linear, and the feasible set corresponds to the solutions of a system of linear equations, we speak of linear programming (LP).

In this thesis, we study a generalization of LP known as semidefinite programming (SDP). More precisely, we consider SDP approaches for solving the stable set and max-cut problems, and some variants thereof. The stable set and max-cut problems are fundamental problems in the field of computer science, and the use of SDP for solving these problems has been widely studied. We propose and investigate solution methods for solving the semidefinite programs that arise in these applications, and provide various theoretical results to better understand these programs. In the rest of this chapter, we provide a brief introduction to SDP, how SDP can be used to solve polynomial optimization problems, and the problems that we will apply SDP to.

## 1.1 An SDP problem and its dual

SDP concerns mathematical optimization problems in which the (real) symmetric matrix variable is restricted to be positive semidefinite (PSD). A symmetric  $n \times n$  matrix  $X$  is said to be PSD if  $v^\top X v \geq 0$  for all  $v \in \mathbb{R}^n$ , written as  $X \succeq 0$ . Another equivalent condition is that the eigenvalues of  $X$  are nonnegative. We write  $\mathcal{S}_+^n$  for the cone of symmetric PSD matrices of size  $n \times n$ , and  $\mathcal{S}^n$  for the set of symmetric matrices of size  $n \times n$ . An SDP problem (or simply, an SDP) can be stated as follows:

$$p^* := \inf_{X \in \mathcal{S}_+^n} \langle C, X \rangle \text{ subject to } \langle A_i, X \rangle = b_i \quad i = 1, \dots, m, \quad (1.1)$$

where  $C, A_1, \dots, A_m \in \mathcal{S}^n$ ,  $b \in \mathbb{R}^m$ , and the trace inner product  $\langle C, X \rangle$  is defined as

$$\langle C, X \rangle := \text{tr}(CX) = \sum_{i,j \in [n]} C_{ij} X_{ij}.$$

Every SDP admits a corresponding dual SDP. The dual SDP of (1.1) is given by

$$d^* := \sup_{y \in \mathbb{R}^m} b^\top y \text{ subject to } C - \sum_{i=1}^m y_i A_i \succeq 0. \quad (1.2)$$

It is common to set  $p^* = \infty$  if (1.1) is infeasible, and  $p^* = -\infty$  if (1.1) is unbounded. Similarly, we set  $d^* = -\infty$  if (1.2) is infeasible, and  $d^* = \infty$  if (1.2) is unbounded. We have that  $d^* \leq p^*$ , which is known as weak duality. Strong duality holds when  $d^* = p^*$ . It is known that strong duality holds if Slater's condition [285] is satisfied by (1.1) or (1.2). Slater's condition of an optimization problem is that it is bounded and strictly feasible.

Strict feasibility of (1.1) and (1.2) involves the notion of a positive definite matrix. A symmetric matrix  $X$  is said to be positive definite, written as  $X \succ 0$ , if its eigenvalues are positive. Then, the SDP (1.1) is said to be strictly feasible if its feasible set contains a positive definite matrix. Similarly, (1.2) is said to be strictly feasible if there exists a  $y \in \mathbb{R}^m$  for which  $C - \sum_{i=1}^m y_i A_i \succ 0$ .

Many excellent surveys and textbooks have been written on SDP, see e.g., [13, 29, 178, 188, 212, 240, 296, 300, 313] and references therein. These references cover the wide range of applications of SDP, and its rich theory. In the rest of this thesis, we limit ourselves to a small number of these topics.

## 1.2 Polynomial optimization

Polynomial optimization concerns optimization problems in which the objective function is a polynomial, and the constraints are given by polynomial (in)equalities. Polynomial optimization has been popularized by the works of Shor [276] and Nesterov [235], and further refined by Lasserre [175] and Parrilo [249]. These works demonstrate that SDP is a powerful tool for solving polynomial optimization problems (POPs). In this thesis, we consider POPs of the following form:

$$f_{\min} := \inf_{x \in K} f(x), \text{ for } K := \{x \in \mathbb{R}^n : g(x) = 0 \text{ for all } g \in S\}, \quad (1.3)$$

where  $f \in \mathbb{R}[x]$ , with  $\mathbb{R}[x]$  the ring of polynomials in the  $n$  variables  $x_1, \dots, x_n$ , and  $S$  is a finite subset of  $\mathbb{R}[x]$ . In general, (1.3) is NP-hard, since it includes, for example, the NP-hard stable set problem (see Section 1.3.1).

Let  $\mathbb{P}(K)$  be the set of polynomials nonnegative over  $K$ . It can be observed that (1.3) is equivalent to

$$f_{\min} = \sup \{\mu \in \mathbb{R} : f - \mu \in \mathbb{P}(K)\}, \quad (1.4)$$

see e.g., [212, Eq. 2.6]. It follows from (1.4) that tractable lower bounds on  $f_{\min}$  can be obtained by replacing  $\mathbb{P}(K)$  in (1.4) by a tractable subset of  $\mathbb{P}(K)$ . One class of tractable subsets of  $\mathbb{P}(K)$  is based on the notion of sum of squares (SOS) polynomials. The set of SOS polynomials of degree at most  $d \in \mathbb{N}$  is given by

$$\Sigma_d := \left\{ p \in \mathbb{R}[x]_d : p = \sum_{i=1}^k p_i(x)^2, \quad k \in \mathbb{N}, p_i \in \mathbb{R}[x] \text{ for all } i \in [k] \right\},$$

where  $\mathbb{R}[x]_d$  denotes the set of polynomials of degree at most  $d$ . SOS polynomials can be expressed using PSD matrices. To see this, define  $\mathbf{v}_d(x)$  as the vector of monomials that form a basis of  $\mathbb{R}[x]_d$ , of size  $s(d) := \binom{n+d}{d}$ . We have that

$$p \in \Sigma_{2d} \iff \exists Q \in \mathcal{S}_+^{s(d)} \text{ satisfying } p = \mathbf{v}_d(x)^\top Q \mathbf{v}_d(x),$$

see e.g., [178, Prop. 2.1]. Clearly, if  $p$  is SOS, then  $p$  is nonnegative over  $\mathbb{R}^n$ , but the converse is not necessarily true [140].

Let  $\mathcal{I}_S$  be the polynomial ideal generated by the  $g \in S$ . The ideal  $\mathcal{I}_S$  and the SOS polynomials can be used to define a subset of  $\mathbb{P}(K)$ , given by

$$\mathcal{M}(S)_d := \left\{ p \in \mathbb{R}[x]_{2d} : p \equiv \mathbf{v}_d(x)^\top Q \mathbf{v}_d(x) \pmod{\mathcal{I}_S}, Q \in \mathcal{S}_+^{s(d)} \right\},$$

for some  $d \in \mathbb{N}$ . Indeed, if  $p \in \mathcal{M}(S)_d$ , then  $p(x) \geq 0$  for all  $x \in K$ , and thus,  $\mathcal{M}(S)_d \subseteq \mathbb{P}(K)$ . Since  $Q \succeq 0$ , SDP can be used to optimize over  $\mathcal{M}(S)_d$ , for fixed  $d \in \mathbb{N}$ .

If we now replace the constraint  $f - \mu \in \mathbb{P}(K)$  in (1.4) by the more restrictive constraint  $f - \mu \in \mathcal{M}(S)_d$ , we obtain the following lower bound on  $f_{\min}$ :

$$f_d := \sup \{ \mu \in \mathbb{R} : f - \mu \in \mathcal{M}(S)_d \}. \quad (1.5)$$

Note that the condition  $f - \mu \in \mathcal{M}(S)_d \subseteq \mathbb{R}[x]_{2d}$  in (1.5) requires that the relaxation order  $d \geq \lceil \deg f/2 \rceil$ . The SDPs defining  $f_d$ , for  $d \geq \lceil \deg f/2 \rceil$ , form a hierarchy of SDP relaxations of the POP (1.3). This hierarchy is known as the Lasserre hierarchy [175], or Moment-SOS hierarchy [179]. Note that the original presentation of the Lasserre hierarchy in [175], involves only polynomial inequality constraints. Presentations of the Lasserre hierarchy with polynomial equality constraints can also be found in, e.g., [120, 174, 184].

The SDP defining  $f_d$  is referred to as the  $d$ th level of the Lasserre hierarchy. Higher levels of the hierarchy provide better bounds on  $f_{\min}$  than lower levels of the hierarchy, since

$$\mathcal{M}(S)_d \subseteq \mathcal{M}(S)_{d+1} \subseteq \mathbb{P}(K) \quad \forall d \in \mathbb{N}$$

implies that  $f_d \leq f_{d+1} \leq f_{\min}$ . At the same time, the PSD matrix variable of the  $d$ th level of the hierarchy is of size  $s(d) \times s(d)$ , and so ‘*the computational cost of its basic formulation can be quite heavy, even for problems of modest dimension*’. [179, Sect. 8].

If, for some  $d \in \mathbb{N}$ ,  $\mathcal{M}(S)_d$  satisfies the Archimedean condition, i.e.,  $R - \|x\|^2 \in \mathcal{M}(S)_d$  for some  $R > 0$ , then  $\lim_{d \rightarrow \infty} f_d = f_{\min}$  [175, Thm. 4.2.a]. There also exist additional conditions under which the hierarchy exhibits finite convergence [66, 177, 184, 238, 239], that is,  $f_d = f_{\min}$  for some finite  $d \in \mathbb{N}$ . Lastly, if the polynomials  $g \in S$  form a Gröbner basis for  $\mathcal{I}_S$ , then for fixed  $d \geq \lceil \deg f/2 \rceil$ , the SDP bound  $f_d$  is computable in time polynomial in  $n$  up to fixed precision [262, Lem. 8]. This condition is satisfied for the sets  $S$  we consider in this thesis. There exist more general conditions that ensure polynomial time computability of  $f_d$  up to fixed precision, see [123, 242, 262].

### 1.2.1 A dual perspective on polynomial optimization

Instead of using SOS polynomials, the Lasserre hierarchy (1.5) can also be stated in terms of truncated moment sequences. This leads to a hierarchy of SDPs that are dual to (1.5), and therefore, provide the same values  $f_d$ , assuming a mild condition that ensures strong duality.

Let us write polynomials in  $\mathbb{R}[x]$  as

$$p(x) = \sum_{\alpha \in \mathbb{N}^n} p_\alpha x^\alpha, \text{ where } p_\alpha \in \mathbb{R} \text{ and } x^\alpha := \prod_{i \in [n]} x_i^{\alpha_i}. \quad (1.6)$$

Given a sequence of real numbers  $\{y_\alpha\}_{\alpha \in \mathbb{N}^n}$  with  $y_0 = 1$ , the associated linear Riesz functional  $L_y : \mathbb{R}[x] \rightarrow \mathbb{R}$  is defined as

$$L_y(p) := \sum_{\alpha \in \mathbb{N}^n} p_\alpha y_\alpha.$$

If  $P$  is a matrix with polynomials as entries, we define  $L_y(P)$  as the matrix satisfying  $(L_y(P))_{ij} = L_y(P_{ij})$ . The moment matrix of order  $d$  is then defined as

$$M_d(y) := L_y(\mathbf{v}_d(x)\mathbf{v}_d(x)^\top).$$

Consider the sets of finite sequences, defined for  $d \geq \max_{g \in S} \deg g/2$ ,

$$\mathcal{Y}_d := \left\{ \{y_\alpha\}_{\alpha \in \mathbb{N}_{2d}^n} : \begin{array}{l} y_0 = 1, M_d(y) \succeq 0, \\ L_y(x^\alpha g(x)) = 0 \ \forall g \in S, \ \forall \alpha \in \mathbb{N}_{2d-\deg(g)}^n \end{array} \right\}, \quad (1.7)$$

where  $S$  is the set of polynomials defining the feasible set  $K$ , see (1.3). The sequences in  $\mathcal{Y}_d$  are known as truncated moment sequences.

The sets  $\mathcal{Y}_d$  are closely related to  $K$ , see (1.3). To see this, let  $z \in K \subseteq \mathbb{R}^n$  and define, for  $\alpha \in \mathbb{N}^n$ , the monomial  $z^\alpha$  as  $x^\alpha$  in (1.6). Consider the sequence  $\{\tilde{y}_\alpha\}_{\alpha \in \mathbb{N}_{2d}^n}$  defined by  $\tilde{y}_\alpha = z^\alpha$  for all  $\alpha \in \mathbb{N}_{2d}^n$ . Clearly,  $\tilde{y}_0 = z^0 = 1$ . Since  $L_{\tilde{y}}(p) = p(z)$  for any  $p \in \mathbb{R}[x]_{2d}$ , we have that the moment matrix satisfies

$$M_d(\tilde{y}) = L_{\tilde{y}}(\mathbf{v}_d(x)\mathbf{v}_d(x)^\top) = \mathbf{v}_d(z)\mathbf{v}_d(z)^\top \succeq 0,$$

Lastly, for any  $g \in S$  and  $\alpha \in \mathbb{N}_{2d-\deg(g)}^n$ , we have that

$$L_{\tilde{y}}(x^\alpha g(x)) = z^\alpha g(z) = z^\alpha \cdot 0 = 0,$$

where  $g(z) = 0$  is implied by  $z \in K$ . Thus,  $\tilde{y} \in \mathcal{Y}_d$ . Stated differently, the map  $x \mapsto \{x^\alpha\}_{\alpha \in \mathbb{N}_{2d}^n}$  defines an injection from  $K$  to  $\mathcal{Y}_d$ . Hence,  $\mathcal{Y}_d$  can be considered as a semidefinite relaxation of the lifted feasible set. That is,  $\{\{x^\alpha\}_{\alpha \in \mathbb{N}_{2d}^n} : x \in K\} \subseteq \mathcal{Y}_d$ .

It follows that the values

$$f_d^* := \inf \{L_y(f) : y \in \mathcal{Y}_d\}, \quad (1.8)$$

which are defined for any integer

$$d \geq \max \left\{ (\deg g/2)_{g \in S}, \lceil \deg f/2 \rceil \right\}, \quad (1.9)$$

satisfy  $f_d^* \leq f_{\min}$  for any such  $d$ . It can be shown [175, Section 4] that the SDPs defining  $f_d^*$  and  $f_d$ , see (1.5), are dual to each other, so that  $f_d^* \geq f_d$  by weak duality. Condition (1.9) suffices for ensuring strong duality [217, Cor. 3.2]. Note that the results from [217] can be applied after transforming our definition of  $K$  to  $\{x \in \mathbb{R}^n : g(x) \geq 0, -g(x) \geq 0 \forall g \in S\}$ .

### 1.2.2 Complex and noncommutative polynomial optimization

The ideas presented in Section 1.2.1 can be adapted for the case of complex polynomials [154, 305], and for the case of noncommutative variables [45, 166, 234, 256]. These adaptations are used in Sections 3.5 and 4.2 respectively.

Let  $\mathbb{C}\langle x \rangle$  be the set of polynomials with complex coefficients, for which the variables  $x_1, \dots, x_n, x_1^*, \dots, x_n^*$  are not assumed to commute. The operator  $*$  is an involution that satisfies  $(x_i)^* = x_i^*$ . A product  $w = w_1 \cdots w_d$  of the variables  $x_1, \dots, x_n, x_1^*, \dots, x_n^*$  is referred to as a word of degree  $d \in \mathbb{N}$ . We define  $w^* := w_d^* w_{d-1}^* \cdots w_1^*$ . For some  $d \in \mathbb{N}$ , let  $\mathcal{W}_d$  be the set of words of degree  $d$  or less, and set  $\mathcal{W} := \mathcal{W}_\infty$ . For a polynomial

$$p(x) = \sum_{w \in \mathcal{W}} p_w w \in \mathbb{C}\langle x \rangle, \quad p_w \in \mathbb{C} \quad \forall w \in \mathcal{W}, \quad (1.10)$$

we define  $p^* := \sum_{w \in \mathcal{W}} \overline{p_w} w^*$ . A polynomial  $p$  is said to be symmetric, or Hermitian, if it satisfies  $p^* = p$ . Given a polynomial  $p$  and a set of square matrices  $X = (X_1, \dots, X_n) \subseteq \mathbb{C}^{d \times d}$ , we define  $p(X) \in \mathbb{C}^{d \times d}$  as the matrix obtained by replacing all  $x_i$  by  $X_i$ , and all  $x_i^*$  by  $X_i^H$ , i.e., the Hermitian transpose of  $X_i$ . Note that for symmetric polynomials  $p$ , the matrix  $p(X)$  is Hermitian.

Given a symmetric polynomial  $f$ , and a set  $S$  of symmetric polynomials, we are interested in solving the following noncommutative POP:

$$\begin{aligned} f_{\min} &:= \inf_{(X,v) \in K_t} v^H f(X) v, \text{ for} \\ K_t &:= \{X = (X_1, \dots, X_n) \subseteq \mathbb{C}^{t \times t}, v \in \mathbb{C}^t : v^H v = 1, g(X) = \mathbf{0} \forall g \in S\}. \end{aligned} \quad (1.11)$$

Note that the dimension  $t \in \mathbb{N}$  is a variable of (1.11). Since  $f$  is symmetric,  $f(X)$  is Hermitian, so that  $v^H f(X) v \in \mathbb{R}$ , and (1.11) is well-defined. An SDP relaxation of (1.11) can be derived in a similar manner as in Section 1.2.1. Let  $(y_w)_{w \in \mathcal{W}}$  be a sequence of complex numbers that satisfies  $y_w = \overline{y_{w^*}}$  for all  $w \in \mathcal{W}$ , and  $y_1 = 1$ . Define the associated linear functional  $L_y : \mathbb{C}\langle x \rangle \rightarrow \mathbb{C}$  as

$$L_y(p) = \sum_{w \in \mathcal{W}} p_w y_w,$$

for polynomials  $p$  as in (1.10). Roughly speaking, the value  $L_y(p)$  models  $v^H p(X) v$  in the relaxation.

Let  $\mathcal{B} \subseteq \mathcal{W}_d$  for some  $d \in \mathbb{N}$ , be a set of words that satisfies  $1 \in \mathcal{B}$ . Define the associated moment matrix as

$$M_{\mathcal{B}}(\mathbf{y}) = (L_y(vw^*))_{v,w \in \mathcal{B}} = (\mathbf{y}_{vw^*})_{v,w \in \mathcal{B}}. \quad (1.12)$$

Note that  $M_{\mathcal{B}}(\mathbf{y})$  is Hermitian due to the fact that  $y_{vw^*} = \overline{y_{wv^*}}$ . Similar as (1.7), we define

$$\mathcal{Y}_{\mathcal{B}} := \left\{ \{y_w\}_{w \in \mathcal{W}_{2d}} : \begin{array}{l} y_1 = 1, M_{\mathcal{B}}(\mathbf{y}) \succeq 0, L_{\mathbf{y}}(vg(x)w) = 0 \\ \forall g \in S, \forall v, w \in \mathcal{W} \text{ s.t. } vg(x)w \in \text{span}(\mathcal{B}) \end{array} \right\}$$

as the set of truncated moment sequences. It can be shown that the value  $f_{\mathcal{B}} := \inf \{L_{\mathbf{y}}(f) : \mathbf{y} \in \mathcal{Y}_{\mathcal{B}}\}$  satisfies  $f_{\mathcal{B}} \leq f_{\min}$ , in a manner similar as we did for (1.8).

### Complex commutative variables on the unit circle

In Section 3.5, we consider the case where the variables  $x_i$  commute, i.e.,  $x_i x_j = x_j x_i$  for all  $i, j \in [n]$ , under the constraints  $x_i^* x_i = 1$  for all  $i \in [n]$ . Since the variables commute, we may assume without loss of generality that  $x_i \in \mathbb{C}$  and  $x_i^* = \overline{x_i}$  for all  $i \in [n]$ , see e.g., [256, Sect. 3.5.2]. Any word  $w$  can thus be written as

$$x^\alpha := \prod_{i \in [n]} x_i^{\alpha_i} \quad \text{for } \alpha \in \mathbb{Z}^n, \quad \text{where } x_i^{\alpha_i} := \begin{cases} x_i^{\alpha_i} & \text{if } \alpha_i > 0 \\ 1 & \text{if } \alpha_i = 0 \\ (\overline{x_i})^{-\alpha_i} & \text{if } \alpha_i < 0. \end{cases}$$

We then consider  $\mathcal{B}$  as a set of vectors in  $\mathbb{Z}^n$ , by identifying  $x^\alpha$  with  $\alpha$ . The associated moment matrix, see (1.12), simplifies as follows, for  $\alpha, \beta \in \mathcal{B} \subseteq \mathbb{Z}^n$ :

$$M_{\mathcal{B}}(\mathbf{y})_{\alpha, \beta} = L_{\mathbf{y}}(x^\alpha (x^\beta)^*) = L_{\mathbf{y}}(x^\alpha \overline{x^\beta}) = L_{\mathbf{y}}(x^\alpha x^{-\beta}) = L_{\mathbf{y}}(x^{\alpha - \beta}) = y_{\alpha - \beta}.$$

In particular, the diagonal entries of  $M_{\mathcal{B}}(\mathbf{y})$  satisfy  $(M_{\mathcal{B}}(\mathbf{y}))_{\alpha, \alpha} = y_{\alpha - \alpha} = y_{\mathbf{0}} = L_{\mathbf{y}}(x^{\mathbf{0}}) = 1$ .

## 1.3 Applications of SDP to selected problems

In this thesis, we are interested in SDP applications for the following problems: the stable set, graph coloring, max-cut and MAX-SAT problems. We briefly outline these problems, as well as some generalizations and variants that we later study in greater detail.

### 1.3.1 Stable sets and graph colorings

Given a graph  $G = (V, E)$ , a subset  $U \subseteq V$  is said to be a stable set in  $G$  if all vertices in  $U$  are pairwise non-adjacent. The stable set problem is to find a stable set in  $G$  of maximum cardinality. The corresponding maximum cardinality is known as the stability number of  $G$ , or independence number, and denoted by  $\alpha(G)$ , i.e.,  $\alpha(G) := \max \{|U| : U \text{ stable set in } G\}$ . The decision problem form of the stable set problem is strongly NP-complete [107].

A (proper) coloring of  $G$  is a mapping  $c : V \rightarrow \mathbb{N}$  that satisfies  $c(i) \neq c(j)$  if  $\{i, j\} \in E$ . The value  $c(i)$  is referred to as the color of vertex  $i$ . The chromatic number of  $G$ , denoted by  $\chi(G)$ , is defined as the smallest number of colors required

to color  $G$ . Stated differently,  $\chi(G)$  is the smallest value  $d \in \mathbb{N}$  for which there exists a coloring  $c : V \rightarrow [d] := \{1, \dots, d\}$  of  $G$ . Like the stability number, also the chromatic number is NP-hard to compute [158].

The first application of SDP to both of these problems is due to Lovász [204], who introduced the Lovász theta number. The Lovász theta number is a graph parameter, denoted by  $\vartheta(G)$ , that can be defined as the optimal value of an SDP, and satisfies the so-called sandwich inequality

$$\alpha(G) \leq \vartheta(G) \leq \chi(\overline{G}). \quad (1.13)$$

The Lovász theta number can be seen as the first level of the Lasserre hierarchy for the stable set problem, see e.g., [120, Sect. 3.1], [184], and [185, Example 8.16]. Let us make this connection explicit.

We first formulate the stable set problem as a POP, see (1.3). A common formulation of  $\alpha(G)$  is the following, where  $n := |V|$ :

$$\alpha(G) = \max \left\{ \sum_{i \in [n]} x_i : \begin{array}{l} x \in \mathbb{R}^n, x_i^2 - x_i = 0 \quad \forall i \in [n], \\ x_i x_j = 0 \quad \forall \{i, j\} \in E \end{array} \right\}. \quad (1.14)$$

Here, the constraints  $x_i^2 - x_i = 0$  enforce that  $x_i \in \{0, 1\}$ . We interpret  $x_i = 1$  as including vertex  $i$  in the stable set. The constraints  $x_i x_j = 0$  for  $\{i, j\} \in E$  then ensure that only one of the adjacent vertices  $i$  and  $j$  are part of the same stable set. We present the first level of the Lasserre hierarchy corresponding to (1.14), from the point of view of truncated moment sequences, as in Section 1.2.1. Thus, the feasible set of the SDP defining  $f_1^*$ , see (1.8), is given by  $\mathcal{Y}_1$ , see (1.7). For  $g(x) = x_i^2 - x_i = 0$ , the corresponding constraints in  $\mathcal{Y}_1$  read

$$M_1(gy) = M_1((x_i^2 - x_i)y) = L_y(x_i^2 - x_i) = y_{2e_i} - y_{e_i} = 0. \quad (1.15)$$

Here,  $e_i \in \{0, 1\}^n$  is the vector that equals 1 at position  $i$ , and 0 elsewhere. It follows from (1.15) that the sequences in  $\mathcal{Y}_1$  depend only on the values  $y_{e_i}$  and  $y_{e_i + e_j}$ ,  $i, j \in V$ . Let us write these values as simply  $y_i$  and  $y_{ij}$  respectively. Now, the constraints  $x_i x_j = 0$  for  $\{i, j\} \in E$  imply  $M_1((x_i x_j)y) = y_{ij} = 0$ . Note additionally that the moment matrix  $M_1(y)$ , indexed by the elements of  $\{\emptyset\} \cup [n]$ , satisfies  $M_1(y)_{\emptyset, \emptyset} = y_{2 \cdot \mathbf{0}} = y_{\mathbf{0}} = 1$ , and  $M_1(y)_{i, i} = y_{2e_i} = y_i = M_1(y)_{\emptyset, i}$ . Lastly, the objective function is given by  $L_y\left(\sum_{i \in [n]} x_i\right) = \sum_{i \in [n]} y_i$ . By writing  $X = M_1(y)$ , the SDP defining  $\vartheta(G)$  can be formulated as

$$\vartheta(G) := \max \left\{ \sum_{i \in [n]} X_{\emptyset, i} : \begin{array}{l} X \in \mathcal{S}_+^{n+1}, X_{ij} = 0 \quad \forall \{i, j\} \in E \\ X_{ii} = X_{\emptyset, i} \quad \forall i \in [n] \end{array} \right\}.$$

The Lovász theta number has been widely studied, see e.g., [51, 103, 127, 130, 167, 206, 221].

### 1.3.1.1 A generalization of $\alpha(G)$ and $\chi(G)$

In Chapter 2, we study, among others, a generalization of  $\vartheta(G)$  that relates to a generalized stable set problem and the problem of  $k$ -multicoloring a graph. Given

$k \in \mathbb{N}$ , a  $k$ -multicoloring of a graph  $G$  is an assignment of  $k$  distinct colors to each  $v \in V$  such that adjacent vertices are assigned disjoint sets of colors [141, 288]. The  $k$ th chromatic number of  $G$ , denoted by  $\chi_k(G)$ , is defined as the least number of colors required to  $k$ -multicolor  $G$ . Similarly,  $\alpha_k(G)$  is defined as the size of a maximum subgraph of  $G$  that can be colored with at most  $k$  colors. That is,

$$\alpha_k(G) := \max_{U \subseteq V(G)} \{|U| : \chi(G[U]) \leq k\},$$

where  $G[U]$  is the vertex-induced subgraph of  $G$  by  $U$ . Note that  $\chi_1(G) = \chi(G)$  and  $\alpha_1(G) = \alpha(G)$ . Narasimhan and Manber [232] introduce the generalized Lovász theta number  $\vartheta_k(G)$ , which they show satisfies a generalization of (1.13), namely

$$\alpha_k(G) \leq \vartheta_k(G) \leq \chi_k(\overline{G}). \quad (1.16)$$

We formally define  $\vartheta_k(G)$  in Chapter 2. In Chapter 2, we compute  $\vartheta_k(G)$  in closed form for various symmetric graphs  $G$ . We also derive various properties of  $\vartheta_k(G)$ , such as that  $\vartheta_k(G)$  is increasing in  $k$ .

### 1.3.2 The max-cut problem

Consider a graph  $G = (V, E)$  with edge weights  $w \in \mathbb{R}^{|E|}$ . A cut of  $G$  is a partition of  $V$  into two disjoint subsets. An edge  $\{i, j\} \in E$  is said to be cut if  $i$  and  $j$  belong to distinct subsets of the partition. The value of a cut is defined as the sum of the weights  $w_e$ ,  $e \in E$ , of the cut edges. The NP-hard max-cut problem is to determine a cut of maximum value, which we denote by  $\text{MC}(G)$ . To model the max-cut problem as a POP, see (1.3), we introduce variables  $x_i \in \{\pm 1\}$ ,  $i \in V$ , that model a partition of  $V$  as  $V = \{i \in V : x_i = 1\} \cup \{i \in V : x_i = -1\}$ . The max-cut problem is then equivalent to computing

$$\text{MC}(G) = \max \left\{ \sum_{\{i,j\} \in E} w_{ij} \frac{1 - x_i x_j}{2} : x \in \mathbb{R}^n, x_i^2 - 1 = 0 \ \forall i \in V \right\}.$$

The value  $\text{MC}(G)$  can also be defined as the solution to an optimization problem over the cut polytope [25], denoted by  $\text{CUT}^n$  and defined as:

$$\text{CUT}^n := \text{Conv} \{xx^\top : x \in \{\pm 1\}^n\}. \quad (1.17)$$

It is not hard to see that

$$\text{MC}(G) = \max_{X \in \text{CUT}^n} \frac{1}{4} \langle L_G, X \rangle, \quad (1.18)$$

where  $L_G$  is the edge-weighted Laplacian matrix of  $G$ , which is the matrix with entries given by

$$(L_G)_{i,j} = \begin{cases} \sum_{k \in N(i)} w_{ik} & \text{if } i = j, \\ -w_{ij} & \text{if } \{i, j\} \in E, \\ 0 & \text{else.} \end{cases}$$

Here,  $\deg(i)$  denotes the degree of vertex  $i$ , i.e., the number of vertices adjacent to  $i$ , and  $N(i)$  denotes the neighborhood of vertex  $i$ . That is,  $N(i) := \{k \in V : \{i, k\} \in E\}$ . A semidefinite relaxation of the max-cut problem can be derived as follows: consider the elliptope [186], defined as

$$\mathcal{E}^n := \{X : X \in \mathcal{S}_+^n, \text{diag}(X) = \mathbf{1}_n\}, \quad (1.19)$$

and note that  $\text{CUT}^n \subseteq \mathcal{E}^n$ . Therefore, it follows from (1.18) that the value

$$\text{MC}_{\text{SDP}}(G) := \max_{X \in \mathcal{E}^n} \frac{1}{4} \langle L_G, X \rangle \quad (1.20)$$

satisfies  $\text{MC}_{\text{SDP}}(G) \geq \text{MC}(G)$ . The relaxation (1.20) was first derived in [71, 72] in its dual form, and further studied in various works, see [181, 183, 186, 258].

Relaxation (1.20) was also used in the famous max-cut approximation algorithm of Goemans and Williamson [117], which achieves an approximation ratio of

$$\alpha_{\text{GW}} := \frac{2}{\pi} \min_{\theta \in [0, \pi]} \frac{\theta}{1 - \cos \theta} \approx 0.878. \quad (1.21)$$

Under the Unique Games Conjecture (UGC) [162], it is NP-hard to approximate the max-cut problem with a ratio greater than  $\alpha_{\text{GW}}$  [163].

**Remark 1.1.** If a problem is NP-hard under the UGC, the problem is said to be UG-hard. If it is UG-hard to approximate a problem with a factor greater than some  $\beta \in \mathbb{R}$ , any  $\beta$ -approximation algorithm is said to be optimal up to UGC.  $\triangle$

### 1.3.2.1 The complex cut polytope

Given integers  $n, m \geq 2$ , we consider the sets

$$\mathcal{U}_m := \{x \in \mathbb{C} : x^m = 1\} \text{ and } \mathcal{U}_m^n := \{x \in \mathbb{C}^n : x_i \in \mathcal{U}_m \forall i \in [n]\}.$$

Note that  $\mathcal{U}_m$  is the set of the  $m$ th roots of unity, and in particular,  $\mathcal{U}_2 = \{\pm 1\}$ . Therefore, the complex cut polytope

$$\text{CUT}_m^n := \text{Conv} \{xx^{\text{H}} : x \in \mathcal{U}_m^n\},$$

where  $(\cdot)^{\text{H}}$  denotes the Hermitian transpose, is a generalization of the cut polytope (1.17), since  $\text{CUT}_2^n = \text{CUT}^n$ . Let

$$\mathcal{H}^n := \{X \in \mathbb{C}^{n \times n} : X_{ij} = \overline{X_{ji}} \text{ for all } i, j \in [n]\}$$

be the set of Hermitian  $n \times n$  matrices, and let  $\mathcal{H}_+^n$  be the set of PSD matrices in  $\mathcal{H}^n$ . It can be observed that  $\text{CUT}_m^n \subseteq \mathcal{H}_+^n$ , which makes  $\text{CUT}_m^n$  well-suited for SDP approximations. Such SDP approximations have been studied in various works [151, 208, 226, 317].

Our analysis of SDP approximations of  $\text{CUT}_m^n$  starts from the following set

$$\mathcal{E}_m^n := \{X : X \in \mathcal{H}_+^n, \text{diag}(X) = \mathbf{1}_n, X_{ij} \in \text{Conv } \mathcal{U}_m \forall i, j \in [n]\},$$

that satisfies  $\text{CUT}_m^n \subseteq \mathcal{E}_m^n$  and  $\mathcal{E}_2^n = \mathcal{E}^n$ , see (1.19). Since for  $m > 2$ ,  $\mathcal{E}_m^n$  also contains matrices with complex entries, we refer to  $\mathcal{E}_m^n$  as the complex elliptope. In Chapter 3, we study the relation between  $\text{CUT}_m^n$  and  $\mathcal{E}_m^n$  in further detail and also consider tighter semidefinite approximations of  $\text{CUT}_m^n$ . We also study the case  $m = \infty$ , with corresponding sets  $\mathcal{U}_\infty := \{x \in \mathbb{C} : |x| = 1\}$  and  $\text{CUT}_\infty^n := \text{Conv}\{xx^H : x \in \mathcal{U}_\infty\}$ .

The sets  $\text{CUT}_m^n$  and  $\mathcal{E}_m^n$  find applications in the multiple-input multiple-output detection problem (MIMO) [151, 208, 226, 317], angular synchronization [24], phase retrieval [303], radar signal processing [209, 287], and  $\text{CUT}_3^n$  can be used to model the max-3-cut problem [118] (a variant of max-cut where the vertices can be partitioned in at most three subsets). For finite  $m \geq 3$ , branch & bound algorithms for optimization over  $\text{CUT}_m^n$  are proposed in [207, 209], in which the bounds are computed using SDP. SDP approximation algorithms for the cases  $m \in \mathbb{N}$  and  $m = \infty$  are studied in [286, 316]. Complex SDP liftings of  $\text{CUT}_\infty^n$ , in the spirit of the Lasserre hierarchy, are also studied in [150].

### 1.3.2.2 The quantum max-cut problem

The quantum max-cut (QMC) problem, and its relation to the max-cut problem, are best understood by first formulating the max-cut problem on some graph  $G$  as an eigenvalue problem. To do so, we use the Pauli matrices, given by

$$X := \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, Y := \begin{bmatrix} 0 & -\mathbf{i} \\ \mathbf{i} & 0 \end{bmatrix}, \text{ and } Z := \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix},$$

where  $\mathbf{i}$  denotes the imaginary unit. For  $n := |V(G)|$  and  $\otimes$  the Kronecker product, we define the extended Pauli matrices

$$\sigma_i := \mathbf{I}_2^{\otimes(i-1)} \otimes \sigma \otimes \mathbf{I}_2^{\otimes(n-i)} \in \mathcal{H}^{2^n}, \text{ for all } \sigma \in \{X, Y, Z\} \text{ and } i \in [n]. \quad (1.22)$$

Let  $\lambda_{\max}(\cdot)$  denote the largest eigenvalue of a given real symmetric or complex Hermitian matrix. It can be shown that

$$\text{MC}(G) = \lambda_{\max}(H_{\text{MC}(G)}), \text{ for } H_{\text{MC}(G)} := \sum_{\{i,j\} \in E(G)} w_{ij} \frac{\mathbf{I}_{2^n} - Z_i Z_j}{2},$$

see, e.g. [112, App. A] or [293, Eq. 6]. The QMC problem on an edge weighted graph  $G = (V, E, w)$  is to compute  $\lambda_{\max}(H_{\text{QMC}(G)})$ , where

$$H_{\text{QMC}(G)} := \sum_{\{i,j\} \in E} w_{ij} H_{ij}, \text{ for } H_{ij} := \mathbf{I}_{2^n} - X_i X_j - Y_i Y_j - Z_i Z_j, \quad (1.23)$$

and the edge weights  $w_e$ ,  $e \in E$ , are restricted to be positive. Note the similarity between the matrices  $H_{\text{QMC}(G)}$  and  $H_{\text{MC}(G)}$ . The matrices  $H_{ij}$  in (1.23) are known as Hamiltonian terms, that act on 2 qubits. The matrix  $H_G := H_{\text{QMC}(G)}$  in (1.23) is known as a 2-local Hamiltonian. The QMC problem is an instance of a  $k$ -local Hamiltonian problem with  $k = 2$ . Moreover, it is QMA-hard [255, Thm. 2], which is the quantum analogue of NP-hard.

The QMC problem can also be considered as a noncommutative POP. The Lasserre hierarchy from Section 1.2 concerns commutative POPs, and is therefore not directly applicable. Instead, we consider SDP relaxations of the QMC problem arising from the NPA hierarchy [234, 256]. Similar SDP relaxations of the QMC problem can also be found in e.g. [112, 164, 190, 191, 247]. We briefly outline the construction of these relaxations, see also Section 1.2.2.

Consider the set of noncommutative variables  $\mathbf{p}^n := \{x_i, y_i, z_i : i \in [n]\}$ , or simply  $\mathbf{p}$  if the context is clear, and let  $\mathbb{C}\langle\mathbf{p}\rangle$  and  $\mathbb{R}\langle\mathbf{p}\rangle$  be the set of polynomials in the variables  $\mathbf{p}$  with complex and real coefficients respectively. The variables in  $\mathbf{p}$  model the (extended) Pauli matrices (1.22), and so they are noncommutative because the matrix multiplication of Pauli matrices is noncommutative. The (anti)commutation relations of the Pauli matrices can be encoded by an appropriate ideal of  $\mathbb{C}\langle\mathbf{p}\rangle$ . This ideal is defined in terms of the following set of polynomials:

$$S := \{x_i^2 - 1, y_i^2 - 1, z_i^2 - 1, x_i y_j - y_j x_i, x_i z_j - z_j x_i, y_i z_j - z_j y_i, \\ x_i y_i + y_i x_i, x_i z_i + z_i x_i, y_i z_i + z_i y_i : i, j \in V, i \neq j\}.$$

Now define the two-sided ideal generated by the polynomials in  $S$ , and truncated at degree  $k$  for some  $k \in \mathbb{N}$ , as

$$\mathcal{I}_k := \left\{ \sum_{f \in S} g_f f h_f : g_f, h_f \in \mathbb{C}\langle\mathbf{p}\rangle, \deg(g_f f h_f) \leq k \forall f \in S \right\}.$$

We define  $\mathcal{I} := \mathcal{I}_\infty$ . For  $f \in \mathbb{C}\langle\mathbf{p}\rangle$ , define  $f(\mathbf{P}) \in \mathcal{H}^{2^n}$  as the matrix obtained by replacing in  $f$  the variables by their corresponding Pauli matrices, see (1.22), and 1 by  $\mathbf{I}_{2^n}$ . Then  $f(\mathbf{P}) = \mathbf{0}$  if and only if  $f \in \mathcal{I}$ . Let  $w$  be a product of variables in  $\mathbf{p}$ . We define  $w^*$  as the monomial obtained by reversing the order of products in  $w$ . The involution  $(\cdot)^*$  is extended to  $\mathbb{C}\langle\mathbf{p}\rangle$  and  $\mathbb{R}\langle\mathbf{p}\rangle$  by conjugate linearity.

Let  $\mathbb{R}\langle\mathbf{p}\rangle_k$  be the set of polynomials in  $\mathbb{R}\langle\mathbf{p}\rangle$  with degree at most  $k$ , for some  $k \in \mathbb{N}$ . Define the dual space  $\mathbb{R}\langle\mathbf{p}\rangle_k^*$  as the set of linear functions  $\mathbb{R}\langle\mathbf{p}\rangle_k \rightarrow \mathbb{R}$ . Define the following subset of  $\mathbb{R}\langle\mathbf{p}\rangle_{2k}^*$ , for some  $k \in \mathbb{N}$ :

$$F_n^k := \left\{ L \in \mathbb{R}\langle\mathbf{p}\rangle_{2k}^* : \begin{array}{l} L(1) = 1, L(f) = 0 \quad \forall f \in \mathbb{R}\langle\mathbf{p}\rangle_{2k} \\ \text{satisfying } f + f^* \in \mathcal{I}_{2k}, M_k(L) \succeq 0 \end{array} \right\}. \quad (1.24)$$

Here,  $n$  is the number of vertices of  $G$ , and  $M_k(L)$  refers to the moment matrix corresponding to  $L$ , similar as in Section 1.2.1. For the sake of conciseness, we do not discuss some of the technicalities pertaining to  $F_n^k$ . A more rigorous presentation is provided in Chapter 4. In particular, Lemma 4.5 proves that any  $L \in F_n^k$  is symmetric, i.e.,  $L(f) = L(f^*)$  for all  $f \in \mathbb{R}\langle\mathbf{p}\rangle_{2k}$ .

Lastly, let  $H_G(\mathbf{p}) \in \mathbb{R}\langle\mathbf{p}\rangle$  be the quadratic polynomial obtained after replacing the Pauli matrices in  $H_G$ , see (1.23), by their corresponding variables, and  $\mathbf{I}_{2^n}$  by 1. We have the following upper bound on  $\lambda_{\max}(H_G)$ , parametrized by some  $k \in \mathbb{N}$ :

$$\max_{L \in F_n^k} L(H_G(\mathbf{p})). \quad (1.25)$$

To see that (1.25) defines an upper bound on  $\lambda_{\max}(H_G)$ , consider the functional  $\tilde{L}(f) := \langle \psi | (f(\mathbf{P}) + f^*(\mathbf{P})) | \psi \rangle / 2$ , where  $\psi$  is a unit length eigenvector of  $H_G$  corresponding to  $\lambda_{\max}(H_G)$ . It can be shown that  $\tilde{L} \in F_n^k$  for any  $k \in \mathbb{N}$ , see (1.24), and that  $\tilde{L}(H_G(\mathbf{p})) = \lambda_{\max}(H_G)$  [190, App. A.1]. In Chapter 4, we use the SDP relaxation (1.25) in three QMC approximation algorithms, for general, triangle-free and bipartite graphs respectively.

QMC approximation algorithms have been widely studied in the literature [18, 112, 145, 153, 156, 164, 190, 191, 246, 247]. The current best known QMC approximation ratio for general graphs equals 0.611 [18], which is far from its conjectured 0.956 upper bound [146]. More specifically, assuming a conjecture involving Gaussian random variables, it is UG-hard (Remark 1.1) to approximate the QMC problem with a factor larger than 0.956. Recall that the famous max-cut approximation algorithm by Goemans and Williamson [117] is optimal up to UGC.

### 1.3.3 The (maximum-)satisfiability problem

Given a logical proposition  $\phi$  on  $n$  boolean variables, the famous satisfiability (SAT) problem is to decide whether there exists a truth assignment to the variables that satisfies  $\phi$ . The SAT problem was the first problem shown to be NP-complete [60]. It is a central problem in mathematical logic and computer science with many applications, see e.g., [19, 109, 160, 216, 225].

Without loss of generality, we may assume that  $\phi$  is in conjunctive normal form (CNF). A proposition  $\phi$  on  $n$  boolean variables  $x_i$  is said to be in CNF if it is written in the following form

$$\phi = \bigwedge_{j=1}^m C_j, \text{ where } C_j = \bigvee_{i \in I_j^+} x_i \bigvee_{i \in I_j^-} \neg x_i \text{ for } I_j^+, I_j^- \subseteq [n]. \quad (1.26)$$

Here, the logical operators  $\wedge$ ,  $\vee$ ,  $\neg$  denote the logical *and*, *or*, and *negation* respectively. The propositions  $C_j$ ,  $j \in [m]$ , are referred to as clauses. It follows that a truth assignment satisfies  $\phi$  if and only if it satisfies all clauses.

The maximum-satisfiability (MAX-SAT) problem is the optimization variant of the SAT problem. Given a proposition  $\phi$  as in (1.26), the MAX-SAT problem is to determine a truth assignment that satisfies the maximum number of clauses. To formulate the MAX-SAT problem as a POP, see (1.3), we require some notation.

With slight abuse of notation, we consider  $C_j$  as both a logical proposition as in (1.26), and a subset of  $[n]$  via  $C_j = I_j^+ \cup I_j^-$ . We refer to  $\ell_j := |C_j|$  as the length of clause  $C_j$ . We associate  $x_i = 1$  with setting  $x_i$  to **true**, and  $x_i = -1$  with setting  $x_i$  to **false**. For each clause  $C_j$ , we define the associated vector  $a_j \in \{0, \pm 1\}^n$  as the vector satisfying

$$a_{j,i} = 1 \text{ if } i \in I_j^+, a_{j,i} = -1 \text{ if } i \in I_j^-, a_{j,i} = 0 \text{ otherwise.}$$

We consider the following polynomial associated to  $\phi$ , see also [117, Sect. 7.2.1],

$$F_\phi(x) := \sum_{j \in [m]} F_\phi^j(x), \text{ where } F_\phi^j(x) := \frac{1}{2^{\ell_j}} \prod_{i \in C_j} (1 - a_{j,i} x_i).$$

Note that for any  $x \in \{\pm 1\}^n$ ,  $F_\phi^j(x) \in \{0, 1\}$ , with  $F_\phi^j(x) = 0$  if and only if  $x$  corresponds to a truth assignment that satisfies clause  $C_j$ . Hence, for any truth assignment  $x \in \{\pm 1\}^n$ , the value  $F_\phi(x)$  equals the number of unsatisfied clauses. Because maximizing the number of satisfied clauses is equivalent to minimizing the number of unsatisfied ones, it follows that the MAX-SAT problem can be formulated as

$$\min_{x \in \{\pm 1\}^n} F_\phi(x). \quad (1.27)$$

Since  $\{\pm 1\}^n = \{x \in \mathbb{R}^n : 1 - x_i^2 = 0 \forall i \in [n]\}$  and  $F_\phi(x)$  is a polynomial, (1.27) defines a POP. In particular, the Lasserre hierarchy can be applied in order to obtain MAX-SAT SDP relaxations. These SDP relaxations are similar to the SDP relaxations we presented for the max-cut problem, see Section 1.3.2, since we modelled both problems on the same feasible set  $\{\pm 1\}^n$ .

The first application of SDP to the MAX-SAT problem is due to Goemans and Williamson [117]. For the MAX-2-SAT problem (MAX- $k$ -SAT instances are MAX-SAT instances that satisfy  $\max_{j \in [m]} \ell_j = k$ ), their algorithm achieves an approximation ratio of  $\alpha_{GW}$ , see (1.21). This ratio was subsequently improved in various works [88, 192, 220] to the ratio  $\beta \approx 0.940$ . It is UG-hard (Remark 1.1) to approximate the MAX-2-SAT problem with a ratio greater than  $\beta$  [35, 163, 229]. Similarly, the MAX-3-SAT problem admits an SDP approximation algorithm with a ratio of  $7/8$  [157] that is optimal if  $P \neq NP$  [135]. An SDP approach to the SAT problem was first proposed in [68], and extended in a series of works by Anjos [9, 10, 11, 12].

In Chapter 6, we consider SDP approaches to the SAT and MAX-SAT problems, from the perspective of SOS polynomials, as outlined in Section 1.2. This approach was also considered in [298], where the resulting SDP relaxations were solved by interior-point methods. We instead solve the relaxations using a variant of the alternating direction method of multipliers (see Section 1.4), and consider significantly larger MAX-SAT instances than those in [298]. We use the SDP relaxations in a branch & bound scheme to solve MAX- $k$ -SAT instances,  $k \in \{2, 3\}$ , from the annual MaxSAT Evaluation, a competition for MAX-SAT solvers.

## 1.4 The ADMM and its variants

The alternating direction method of multipliers (ADMM) is a first-order method for solving convex optimization problems. In this thesis (specifically, Chapters 5 and 6), we consider the ADMM only for solving SDPs, which are a special case of convex optimization problems. For a more general treatment of the ADMM, see [34].

Many authors in recent years have proposed the ADMM (or its variants) for solving SDPs, see [46, 69, 70, 121, 196, 211, 312]. Compared to the interior-point method (IPM), which is the classical and standard method for solving SDPs, the ADMM provides less accurate solutions. However, for SDPs with large matrix variables, the ADMM can be much faster than the IPM. Furthermore, the IPM requires much more memory than the ADMM, in some cases requiring even more memory than what is available on the hardware that is used to run these algorithms.

To apply the ADMM to a general SDP (1.1), we first reformulate (1.1) as the following optimization problem, where  $\mathcal{A}Y := (\langle A_1, Y \rangle, \dots, \langle A_m, Y \rangle)$ :

$$\begin{aligned} \inf_{X, Y \in \mathcal{S}^n} \langle C, Y \rangle \\ \text{s.t. } X \in \mathcal{S}_+^n, Y \in \mathcal{F} := \{Y \in \mathcal{S}^n : \mathcal{A}Y = b\}, X = Y. \end{aligned} \quad (1.28)$$

Given a penalty parameter  $\rho > 0$ , the augmented Lagrangian associated to (1.28), with respect to the constraint  $X = Y$ , is the function

$$\mathcal{L}_\rho(X, Y, Z) := \langle C, Y \rangle + \langle Z, Y - X \rangle + \frac{\rho}{2} \|Y - X\|^2,$$

where  $Z \in \mathcal{S}^n$  is the dual variable, and the Frobenius norm  $\|M\| := \sqrt{\text{tr}(MM)}$  for a symmetric matrix  $M \in \mathcal{S}^n$ . Given some initial  $X^1, Y^1, Z^1 \in \mathcal{S}^n$ , we consider the sequence  $(X^\ell, Y^\ell, Z^\ell)_{\ell \in \mathbb{N}}$ , defined recursively as

$$\begin{aligned} X^{\ell+1} &:= \arg \min_{X \succeq 0} \mathcal{L}_\rho(X, Y^\ell, Z^\ell) = \mathcal{P}_{\mathcal{S}_+^n} \left( Y^\ell + \frac{1}{\rho} Z^\ell \right) \\ Z^{\ell+\frac{1}{2}} &:= Z^\ell + \rho \nu_1 (Y^\ell - X^{\ell+1}) \\ Y^{\ell+1} &:= \arg \min_{Y \in \mathcal{F}} \mathcal{L}_\rho(X^{\ell+1}, Y, Z^{\ell+\frac{1}{2}}) = \mathcal{P}_{\mathcal{F}} \left( X^{\ell+1} - \frac{1}{\rho} C - \frac{1}{\rho} Z^{\ell+\frac{1}{2}} \right) \\ Z^{\ell+1} &:= Z^{\ell+\frac{1}{2}} + \rho \nu_2 (Y^{\ell+1} - X^{\ell+1}), \end{aligned} \quad (1.29)$$

where  $\nu_1, \nu_2 \in \mathbb{R}$  are stepsize parameters, and  $\mathcal{P}_{\mathcal{S}_+^n}, \mathcal{P}_{\mathcal{F}}$  denote the orthogonal projection on the sets  $\mathcal{S}_+^n$  and  $\mathcal{F}$  respectively. That is, for some  $M \in \mathcal{S}^n$ ,

$$\mathcal{P}_{\mathcal{S}_+^n}(M) := \arg \min_{X \in \mathcal{S}_+^n} \|X - M\| \quad \text{and} \quad \mathcal{P}_{\mathcal{F}}(M) := \arg \min_{X \in \mathcal{F}} \|X - M\|.$$

To derive the second and fifth equalities in (1.29), see e.g., equations (3.4) and (3.5) in [243]. The scheme (1.29) encompasses the ADMM and many of its variants, known under different names in the literature. The original ADMM is the scheme (1.29) with  $\nu_1 = 0$ . When  $\nu_1, \nu_2 \neq 0$ , (1.29) is known as the Peaceman-Rachford splitting method (PRSM) [251], or symmetric ADMM. It is worth noting that (1.29) corresponds to the unscaled form of the PRSM, which is slightly less efficient than its scaled form, see Appendix A.1 on Page 191.

The matrices  $X^\ell$  and  $Y^\ell$  converge with rate  $\mathcal{O}(1/\ell)$  (in the ergodic sense) to an optimal solution to (1.28) when  $(\nu_1, \nu_2) \in \mathcal{D}$ , where

$$\mathcal{D} := \left\{ (\nu_1, \nu_2) \in \mathbb{R}^2 : \begin{array}{l} |\nu_1| < \min\{1, 1 + \nu_2 - \nu_2^2\}, \\ 0 < \nu_2 < \frac{1+\sqrt{5}}{2}, \nu_1 + \nu_2 > 0 \end{array} \right\},$$

see [136, Thm. 6.5].

We briefly elaborate on how to compute the two projections in (1.29). For any  $X \in \mathcal{S}^n$ , with eigendecomposition  $X = \sum_{i \in [n]} \lambda_i u_i u_i^\top$ ,  $u_i^\top u_i = 1$  for all  $i \in [n]$ , it is well known that

$$\mathcal{P}_{\mathcal{S}_+^n}(X) = \sum_{i \in [n]: \lambda_i > 0} \lambda_i u_i u_i^\top, \quad (1.30)$$

see e.g., [33, Page 399]. The standard method of computing (1.30) is to compute the full eigendecomposition of  $X$ . Given such an eigendecomposition, if the number of negative eigenvalues is sufficiently small, it is more efficient to compute (1.30) as  $\mathcal{P}_{\mathcal{S}_+^n}(X) = X - \sum_{i:\lambda_i < 0} \lambda_i u_i u_i^\top$ . Regardless, the computation of the eigendecompositions often forms the bottleneck of (1.29). This has motivated research into (approximately) computing  $\mathcal{P}_{\mathcal{S}_+^n}$  with methods that avoid computing the full eigendecomposition [152, 266].

To compute a projection of a matrix onto  $\mathcal{F}$ , see (1.28), we make the following observations. Let  $\tilde{n} := n(n+1)/2$ . Observe that  $\mathcal{S}^n$  is isomorphic to  $\mathbb{R}^{\tilde{n}}$ , since  $\text{vech}(\mathcal{S}^n) = \mathbb{R}^{\tilde{n}}$ . Here,  $\text{vech}(\cdot)$  denotes the half-vectorization of its input matrix, that is, for  $X \in \mathcal{S}^n$ ,  $\text{vech}(X) \in \mathbb{R}^{\tilde{n}}$  is the vector of the  $\tilde{n}$  lower triangular entries of  $X$ . Hence, for  $X, Y \in \mathcal{S}^n$ , we have that  $\langle X, Y \rangle = \text{vech}(X)^\top D \text{vech}(Y)$ . Here,  $D$  is a diagonal matrix with  $\text{diag}(D) \in \{1, 2\}^{\tilde{n}}$ , where the values of 2 account for the off-diagonal elements of  $X$  and  $Y$  that are counted twice in the product  $\langle X, Y \rangle$ . It follows that  $\mathcal{F}$  is isomorphic to some affine space in  $\mathbb{R}^{\tilde{n}}$ . We denote this affine space by  $\mathcal{W} := \{x \in \mathbb{R}^{\tilde{n}} : WD^{1/2}x = b\}$ , where  $W \in \mathbb{R}^{m \times \tilde{n}}$  is a matrix such that  $AX = WD^{1/2}\text{vech}(X)$  for all  $X \in \mathcal{S}^n$ . Let us now consider the projection of some matrix  $Z$  onto  $\mathcal{F}$ , where we write  $z = \text{vech}(Z)$ , and  $z' = D^{1/2}z$ . This yields

$$\begin{aligned} \min_{X \in \mathcal{F}} \|X - Z\| &= \min_{x \in \mathcal{W}} (x - z)^\top D (x - z) = \min_{x \in \mathcal{W}} \left\| D^{1/2} (x - z) \right\|^2 \\ &= \min_{y: Wy=b} \|y - z'\|^2 = \min_{y: Wy=b} \|y - z'\|^2. \end{aligned} \quad (1.31)$$

For the third equality, we have substituted  $y = D^{1/2}x$ . The last expression in (1.31) is the problem of projecting  $z'$  onto an affine space, which has a well-known solution, see e.g., [257, Eq. 1], given by

$$\arg \min_{y: Wy=b} \|y - z'\|^2 = z' - W^\top (WW^\top)^{-1} (Wz' - b). \quad (1.32)$$

It is often recommended, for numerical stability, to compute  $(WW^\top)^{-1}(Wz' - b)$  by solving the linear system  $WW^\top x = Wz' - b$  for  $x$  instead. Since the ADMM requires computing (1.32) at every iteration, it is then efficient to compute, for example, a Cholesky decomposition of  $WW^\top$  that can be reused at every iteration. Alternatively, the linear system can be solved by iterative methods [79, 119, 257].

However, in all the cases considered in this thesis,  $WW^\top$  will be a diagonal matrix, which makes  $(WW^\top)^{-1}$  easy to compute. In these cases, using (1.32) is the preferred method for solving (1.31). Only in Section 6.8.1 do we consider a PRSM with additional variables, in which (1.32) is not applicable. In particular, for this PRSM, minimization of the augmented Lagrangian  $\mathcal{L}_\rho$  over  $\mathcal{F}$  does not reduce to computing a projection onto  $\mathcal{F}$ , as it does in (1.29).

## 1.5 Overview of the thesis

We provide a brief overview of the following self-contained chapters in this thesis.

**Chapter 2: The generalized  $\vartheta$ -number and related problems for highly symmetric graphs.** We study the multichromatic number  $\chi_k(G)$  (see Sect. 1.3.1.1) for various highly symmetric graphs. The generalized Lovász theta number  $\vartheta_k$  can be computed by solving an SDP, and provides a bound on  $\chi_k(G)$ , see (1.16). This chapter provides a closed-form expression for  $\vartheta_k(G)$  for several families of graphs such as strongly regular graphs, Kneser graphs and Johnson graphs. Several properties of  $\vartheta_k$  are provided. In particular, it is shown that for any  $\varepsilon > 0$ , there exists a graph  $G$  and integer  $k$  such that  $0 < \vartheta_{k+1}(G) - \vartheta_k(G) < \varepsilon$ . Nordhaus-Gaddum [241] type results on the parameter  $\chi_k(G)$  are presented, i.e., tight lower and upper bounds on  $\chi_k(G)\chi_k(\overline{G})$  and  $\chi_k(G) + \chi_k(\overline{G})$  in terms of  $k$  and  $|V(G)|$ , where  $\overline{G}$  is the complement graph of  $G$ .

**Chapter 3: Cuts and semidefinite liftings for the complex cut polytope.** We study the complex cut polytope, a generalization of the well-known real cut polytope (see Sect. 1.3.2.1). Semidefinite liftings and facet defining inequalities of the complex cut polytope are provided, and these are used to obtain strong approximations of the complex cut polytope. We investigate the use of these facet defining inequalities and liftings for solving MIMO and angular synchronization problems. These two problems arise in wireless communications, and involve retrieving a signal given noisy observations. Both problems can be modelled using the complex cut polytope. Numerical results show that the facet defining inequalities and liftings significantly improve the strength of the SDP relaxations of MIMO and angular synchronization. On the theoretical side, we provide various results on the quality of these semidefinite approximations. For the set  $\text{CUT}_\infty^n$ , we reconsider some of its semidefinite liftings that have been proposed in the literature. It is shown that these liftings can be computed more efficiently by reducing the size of the matrix variable, without weakening the approximation of  $\text{CUT}_\infty^n$ .

This chapter is based on the paper [283] (see also Table 1.1). During the compilation of this thesis, we realized that the proofs of [283, Thm. 4, Lem. 11] were incorrect. We have replaced these results with weaker ones. Specifically, Lemma 3.33 and Theorem 3.34 provide weaker versions of [283, Thm. 4], and Lemma 3.35 provides a weaker version of [283, Lem. 11].

**Chapter 4: Improved approximation ratios for the quantum max-cut problem on general, triangle-free and bipartite graphs.** We consider the QMA-hard QMC problem, see Sect. 1.3.2.2. In particular, we consider three QMC approximation algorithms, for general, triangle-free and bipartite graphs respectively. We prove that our approximation algorithms for triangle-free and bipartite graphs attain the current best-known approximation ratios for their respective graph classes. Our analysis of the QMC algorithm for general graphs has been used in [18], to prove that their QMC algorithm approximation ratio of 0.611, which is the current best-known approximation ratio for general graphs. All the three approximation algorithms in Chapter 4 use an SDP relaxation of the QMC problem. Because the variables underlying these relaxations are noncommutative, the Lasserre hierarchy from Section 1.2 is not applicable. Instead, we resort to the noncommutative variant of the Lasserre hierarchy, known as the NPA hierarchy [234, 256].

Chapter 4 is based on the paper [124] (see also Table 1.1). After posting [124] on arXiv, the paper [18] improved upon [124] in some aspects. Some of the text in [124] has been adjusted in Chapter 4 as a consequence.

**Chapter 5: SDP bounds on the stability number via ADMM and intermediate levels of the Lasserre hierarchy.** We use the ADMM (see Sect. 1.4) to compute bounds from the Lasserre hierarchy for the stable set problem, at levels intermediate to 1 and 2 (and for some graphs, the full level 2). A method is provided to construct an appropriate basis of variables to form these intermediate level relaxations, based on the Lovász theta number. For most of the tested graphs, we compute intermediate levels of the Lasserre hierarchy defined by an SDP wherein the matrix variable is of size 2500. Solving these SDPs with the classical IPM is intractable due to the large memory requirement of the IPM. We therefore solve these SDPs with the ADMM instead. Our approach is shown to be capable of computing strong bounds on the stability number of graphs on at most 300 vertices and 11427 edges. Extensive numerical results compare our approach with other SDP based bounds on the stability number from the literature.

**Chapter 6: On solving the MAX-SAT problem using sum of squares.** We use SOS polynomials and the Lasserre hierarchy, see Section 1.2, to compute SDP bounds on the NP-hard MAX-SAT problem, see Section 1.3.3. These bounds are computed by using levels of the Lasserre hierarchy that are intermediate to levels 1 and 2. The corresponding SDPs are solved with an ADMM variant known as the Peaceman-Rachford splitting method, see Section 1.4. We present a MAX-SAT solver that utilizes these SDP bounds in a branch & bound scheme. Our solver is benchmarked on instances from the annual MAX-SAT competition for MAX-SAT solvers, and shown to be competitive with the other solvers in this competition. We also reconsider SDP approaches to the SAT problem from the literature. In particular, a new SDP relaxation of the SAT problem is considered. It is shown that this relaxation is exact if the rank of the matrix variable is constrained to be at most 2.

This chapter is based on the paper [282] (see also Table 1.1). During the compilation of this thesis, we realized that the proof of [282, Thm. 1] was incorrect. We have replaced [282, Thm. 1] with Theorem 6.1. The statement of Theorem 6.1 is similar to [282, Thm. 1], but uses a different basis of monomials.

## 1.6 Contributions to the literature

Each chapter of this thesis (except for Chapter 1) is based on a corresponding paper, see Table 1.1. These chapters are mostly identical to their corresponding paper, apart from the corrections outlined in Section 1.5.

The author of this thesis has also co-authored the paper:

A. Ghaffari-Hadigheh, L. Sinjorgo, and R. Sotirov. On convergence of a  $q$ -random

Ch.	Corresponding paper	cit.
2	L. Sinjorgo and R. Sotirov. On the generalized $\vartheta$ -number and related problems for highly symmetric graphs. <i>SIAM Journal on Optimization</i> , 32(2):1344–1378, 2022	[281]
3	L. Sinjorgo, R. Sotirov, and M. F. Anjos. Cuts and semidefinite liftings for the complex cut polytope. <i>Mathematical Programming</i> , pages 1–50, 2024	[283]
4	S. Gribling, L. Sinjorgo, and R. Sotirov. Improved approximation ratios for the quantum max-cut problem on general, triangle-free and bipartite graphs. <i>preprint arXiv:2504.11120</i> , 2025	[124]
5	L. Sinjorgo, R. Sotirov, and J. C. Vera. SDP bounds on the stability number via ADMM and intermediate levels of the Lasserre hierarchy. <i>preprint arXiv:2506.08648</i> , 2025	[284]
6	L. Sinjorgo and R. Sotirov. On solving MAX-SAT using sum of squares. <i>INFORMS Journal on Computing</i> , 36(2):417–433, 2024	[282]

Table 1.1: Chapters of this thesis and their corresponding paper upon which they are based

coordinate constrained algorithm for non-convex problems. *Journal of Global Optimization*, 90(4):843–868, 2024

which is not included in this thesis.

## 2 The generalized $\vartheta$ -number and related problems for highly symmetric graphs

A  $k$ -multicoloring of a graph is an assignment of  $k$  distinct colors to each vertex in the graph such that two adjacent vertices are assigned disjoint sets of colors. The  $k$ -multicoloring is also known as  $k$ -fold coloring,  $k$ -tuple coloring or simply multicoloring. We denote by  $\chi_k(G)$  the minimum number of colors needed for a  $k$ -multicoloring of a graph  $G$ , and refer to it as the  $k$ th chromatic number of  $G$  or the multichromatic number of  $G$ . Multicoloring seems to have been independently introduced by Hilton et al. [141] and Stahl [288]. The  $k$ -multicoloring is a generalization of the well-known standard graph coloring. Namely,  $\chi(G) := \chi_1(G)$  is known as the chromatic number of a graph  $G$ . Not surprisingly, multicoloring finds applications in comparable areas, such as job scheduling [106, 133], channel assignment in cellular networks [233] and register allocation in computers [50]. There exist several results on  $\chi_k(G)$  for specific classes of graphs. In particular, Lin [197] and Lin et al. [198] consider multicoloring the Mycielskian of graphs, Ren and Bu [263] study multicoloring of planar graphs while Marx [219] proves that the multicoloring problem is strongly NP-hard in binary trees. Cranston and Rabern [61] show that, for any planar graph  $G$ ,  $\chi_2(G) \leq 9$ . This result is implied by the famous four-color theorem [17], but it has a much simpler proof.

The maximum  $k$ -colorable subgraph (MkCS) problem is to find the largest induced subgraph in a given graph that can be colored with  $k$  colors so that no two adjacent vertices have the same color. When  $k = 1$ , the MkCS problem reduces to the well known stable set problem. The MkCS problem is one of the NP-complete problems considered in [193]. We denote by  $\alpha_k(G)$  the number of vertices in a maximum  $k$ -colorable subgraph of  $G$ , and by  $\omega_k(G)$  the size of the largest induced subgraph that can be partitioned into  $k$  cliques. When  $k = 1$ , the graph parameter  $\omega(G) := \omega_1(G)$  is known as the clique number of a graph, and  $\alpha(G) := \alpha_1(G)$  as the independence number of a graph. We note that  $\alpha_k(G) = \omega_k(\overline{G})$ , where  $\overline{G}$  denotes the complement graph of  $G$ . The MkCS problem has a number of applications such as channel assignment in spectrum sharing networks [169, 291], VLSI design [215] and human genetic research [91, 200]. There exist several results on  $\alpha_k(G)$  for specific classes of graphs. The size of the maximum  $k$ -colorable subgraph for the Kneser graph  $K(v, 2)$  is provided by Füredi and Frankl [96]. Yannakakis and Gavril [315] consider the

MkCS problem for chordal graphs, Addario-Berry et al. [4] study the problem for an  $i$ -triangulated graph, and Narasimhan [231] computes  $\alpha_k(G)$  for circular-arc graphs and tolerance graphs.

Narasimhan and Manber [232] introduce a graph parameter  $\vartheta_k(G)$  that serves as a bound on both the minimum number of colors needed for a  $k$ -multicoloring of a graph  $G$  and the number of vertices in a maximum  $k$ -colorable subgraph of  $G$ . The parameter  $\vartheta_k(G)$  generalizes the concept of the famous  $\vartheta$ -number that was introduced by Lovász [204] for bounding the Shannon capacity of a graph [273]. The Lovász theta number is a widely studied graph parameter see e.g., [51, 127, 130, 167, 206, 221]. The Lovász theta number provides bounds for both the clique number and the chromatic number of a graph, both of which are NP-hard to compute. The well-known result that establishes the relation  $\alpha_1(G) \leq \vartheta_1(G) \leq \chi_1(\overline{G})$ , or equivalently  $\omega_1(G) \leq \vartheta_1(\overline{G}) \leq \chi_1(G)$ , is known as the sandwich theorem [206]. The Lovász theta number can be computed in polynomial time as a semidefinite programming (SDP) problem by using interior-point methods. Thus, when the clique number and chromatic number of a graph coincide, i.e., when the graph is weakly perfect, the Lovász theta number provides those quantities in polynomial time.

Despite the popularity of the Lovász theta number, the generalized theta number  $\vartheta_k(G)$  has received little attention in the literature. Narasimhan and Manber [232] show that  $\alpha_k(G) \leq \vartheta_k(G) \leq \chi_k(\overline{G})$  or equivalently  $\omega_k(G) \leq \vartheta_k(\overline{G}) \leq \chi_k(G)$ . These inequalities can be seen as a generalization of the Lovász sandwich theorem. Alizadeh [7] formulates the generalized  $\vartheta$ -number using semidefinite programming. Kuryatnikova et al. [172] introduce the generalized  $\vartheta'$ -number that is obtained by adding nonnegativity constraints to the SDP formulation of the  $\vartheta_k$ -number. The generalized  $\vartheta$ -number and  $\vartheta'$ -number are evaluated numerically as upper bounds for the MkCS problem in [172]. The authors of [172] characterize a family of graphs for which  $\vartheta_k(G)$  and  $\vartheta'_k(G)$  provide tight bounds for  $\alpha_k(G)$ . Here, we study also a relation between  $\vartheta_k(G)$  and  $\chi_k(G)$ , and extend many known results for the Lovász  $\vartheta$ -number to the generalized  $\vartheta$ -number.

## Main results and outline

This chapter provides various theoretical results for  $\alpha_k(G)$ ,  $\vartheta_k(G)$  and  $\chi_k(G)$ . We show numerous properties of  $\vartheta_k(G)$  including results on different graph products of two graphs such as the Cartesian product, strong product and disjunction product. We show that the sequence  $(\vartheta_k(G))_k$  is increasing towards the number of vertices in  $G$ , and that the increments of the sequence can be arbitrarily small. The latter result is proven by constructing a particular graph that satisfies the desired property. We also provide a closed form expression on the generalized  $\vartheta$ -number for several graph classes including complete graphs, complete multipartite graphs, cycle graphs, circulant graphs, strongly regular graphs, orthogonality graphs, the Kneser graphs and some Johnson graphs. Our results show that  $\vartheta_k(G) = k\vartheta(G)$  for the Kneser graphs and more general Johnson graphs, strongly regular graphs, cycle graphs and circulant graphs. This chapter presents lower bounds on the  $k$ th chromatic number for all regular graphs, but also specialized bounds for the Hamming, Johnson and orthogonality graphs. We also provide bounds on the product and sum of  $\chi_k(G)$

and  $\chi_k(\overline{G})$ , and present graphs for which those bounds are attained. Those results generalize well known results of Nordhaus and Gaddum [241] for  $\chi(G)$  and  $\chi(\overline{G})$ .

This chapter is organized as follows. Preliminaries are provided in Section 2.1. In Section 2.2 we formally introduce  $\vartheta_k(G)$  and  $\chi_k(G)$  and show how those graph parameters relate. In Section 2.3 we study the sequence  $(\vartheta_k(G))_k$ . Section 2.4 provides bounds for  $\vartheta_k(G)$  when  $G$  is the strong graph product of two graphs and the disjunction product of two graphs. Section 2.5 provides the values of the generalized  $\vartheta$ -number for complete graphs, cycle graphs, circulant graphs and complete multipartite graphs. In Section 2.5.1 we provide a closed form expression for the generalized  $\vartheta$ -number on the Kneser graphs, as well as for the Johnson graphs. Section 2.5.2 relates  $\vartheta(K_k \square G)$  and  $\vartheta_k(G)$ . We provide a closed form expression for the generalized  $\vartheta$ -number for strongly regular graphs in Section 2.6. In the same section we also relate the Schrijver's number  $\vartheta'(K_k \square G)$  with  $\vartheta_k(G)$  when  $G$  is a strongly regular graph. In Section 2.7 we study a relation between the orthogonality graphs and here considered graph parameters. Section 2.8 provides new lower bounds on the  $k$ th chromatic number for regular graphs and triangular graphs. We present several results for the multichromatic number of the Hamming graphs in Section 2.8.1.

## 2.1 Preliminaries

We denote cycle graphs on  $n$  vertices by  $C_n$ , complete graphs on  $n$  vertices by  $K_n$ , and complete multipartite graph by  $K_{m_1, \dots, m_p}$ . Note that  $K_{m_1, \dots, m_p}$  is a graph on  $\sum_{i=1}^p m_i$  vertices.

**Definition 2.1** (Graph products). An arbitrary graph product of graphs  $G_1 = (V_1, E_1)$  and  $G_2 = (V_2, E_2)$  is denoted by  $G_1 * G_2$ , having as vertex set the Cartesian product  $V_1 \times V_2$ . Table 2.1 shows when vertices  $(v_1, v_2)$  and  $(u_1, u_2)$  are adjacent in  $G_1 * G_2$ , for the lexicographic, tensor, Cartesian, strong and disjunction [1] graph products.

Graph prod.	$G_1 * G_2$	Condition for $\{(v_1, v_2), (u_1, u_2)\} \in E(G_1 * G_2)$
Lexicographic	$G_1 \circ G_2$	$\{v_1, u_1\} \in E_1$ or $[v_1 = u_1$ and $\{v_2, u_2\} \in E_2]$
Tensor	$G_1 \otimes G_2$	$\{v_1, u_1\} \in E_1$ and $\{v_2, u_2\} \in E_2$
Cartesian	$G_1 \square G_2$	$[v_1 = u_1$ and $\{v_2, u_2\} \in E_2]$ or $[v_2 = u_2$ and $\{v_1, u_1\} \in E_1]$
Strong	$G_1 \boxtimes G_2$	$\{(v_1, v_2), (u_1, u_2)\} \in E(G_1 \square G_2) \cup E(G_1 \otimes G_2)$
Disjunction	$G_1 \vee G_2$	$\{v_1, u_1\} \in E_1$ or $\{v_2, u_2\} \in E_2$

Table 2.1: Graph products. Note that each ‘or’ in this table is inclusive.

Graph products are also illustrated on Page 22.

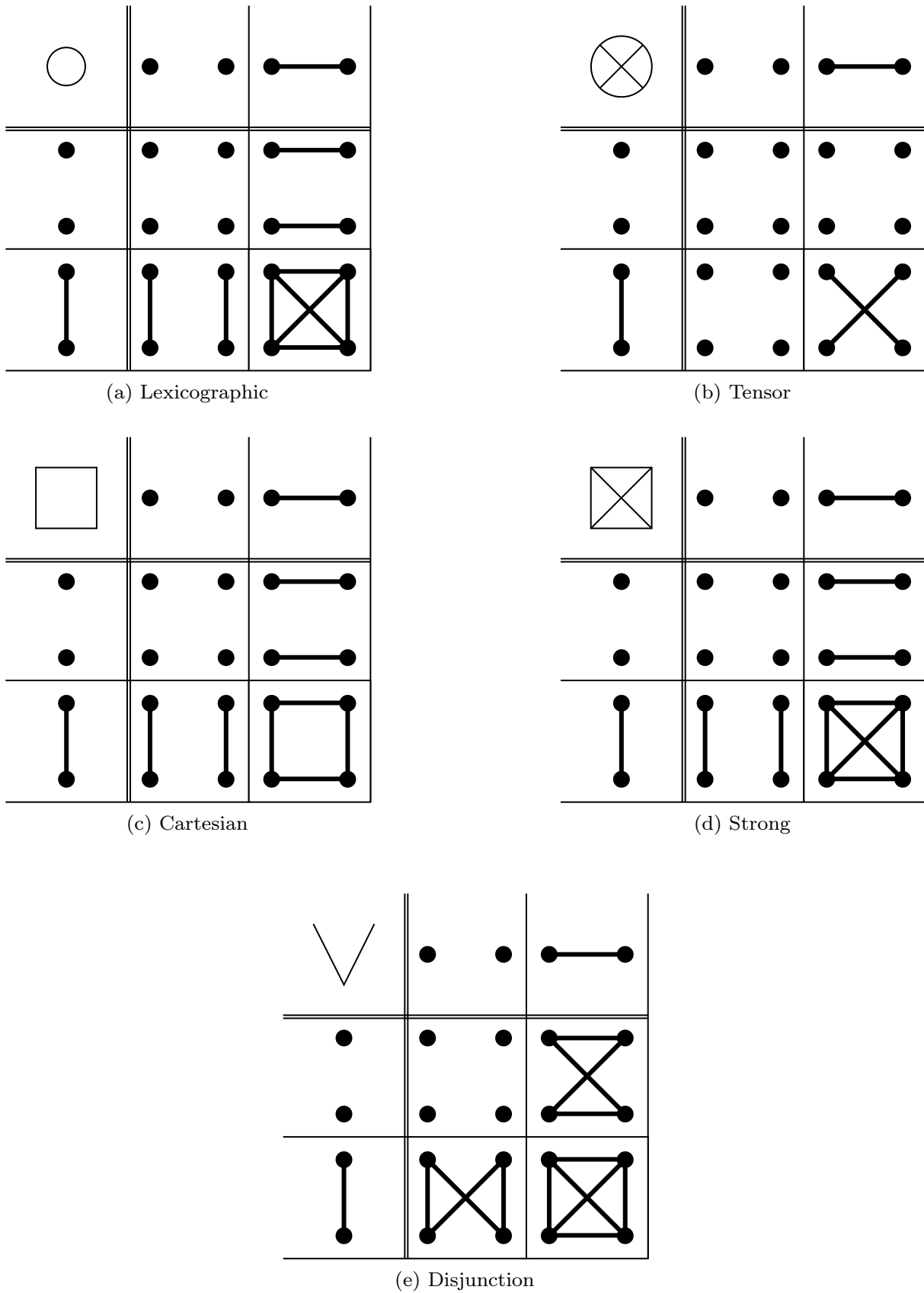


Figure 2.1: Illustration of the graph products from Table 2.1.

In order to define the Hamming graphs, we first provide the definition of the Hamming distance.

**Definition 2.2** (Hamming distance). For two integer valued vectors  $\mathbf{u}$  and  $\mathbf{v}$ , the Hamming distance between them, denoted by  $d(\mathbf{u}, \mathbf{v})$ , is the number of positions in which their entries differ.

**Definition 2.3** (Hamming graph). The Hamming graph  $H(n, q, F)$  for  $n, q \in \mathbb{N}$  and  $F \subseteq \mathbb{N}$  has as vertices all the unique elements in  $(\mathbb{Z}/q\mathbb{Z})^n$ . In the Hamming graph, vertices  $u$  and  $v$  are adjacent if their Hamming distance  $d(u, v) \in F$ .

Many authors define the Hamming graphs only for  $F = \{1\}$ .

**Definition 2.4** (Johnson graph). Let  $n, m \in \mathbb{N}$ ,  $1 \leq m \leq n/2$  and  $f \in \{0, 1, \dots, m\}$ . The Johnson graph  $J(n, m, f)$  has as vertices all the possible  $m$ -sized subsets of  $[n]$ . Denote the subset corresponding to a vertex  $u$  by  $s(u)$ . Then  $|s(u)| = m$  and vertices  $u$  and  $v$  are adjacent if and only if  $|s(u) \cap s(v)| = f$ .

Many authors define the Johnson graph only for  $f = m - 1$ . When  $f = 0$ , the Johnson graph is better known as the Kneser graph.

**Definition 2.5** (Kneser graph). Let  $n, m \in \mathbb{N}$  and  $1 \leq m \leq n/2$ . Then the Kneser graph  $K(n, m)$  is the Johnson graph  $J(n, m, 0)$ .

**Definition 2.6** (Strongly regular graph). A  $d$ -regular graph  $G$  on  $n$  vertices is called strongly regular with parameters  $(n, d, \lambda, \mu)$  if any two adjacent vertices share  $\lambda$  common neighbors and any two non-adjacent vertices share  $\mu$  common neighbors. Alternatively, we say that  $G$  is an  $\text{srg}(n, d, \lambda, \mu)$ .

## 2.2 $\vartheta_k(G)$ and $\chi_k(G)$ formulations and their relation

In this section, we formally introduce the multichromatic number and the generalized  $\vartheta$ -number of a graph. We also show a relationship between these two graph parameters.

Let  $G = (V, E)$  be a simple undirected graph with  $n$  vertices. A  $k$ -multicoloring of  $G$  that uses  $R$  colors is a mapping  $f : V \rightarrow 2^R$ , such that  $|f(i)| = k$  for all vertices  $i \in V$  and  $|f(i) \cap f(j)| = 0$  for all edges  $\{i, j\} \in E$ . The multichromatic number  $\chi_k(G)$  is defined as the smallest  $R$  such that a  $k$ -multicoloring of  $G$  that uses  $R$  colors exists.

Multicoloring can be reduced to standard graph coloring by use of the lexicographic product of graphs, see Definition 2.1. Namely, Stahl [288] showed that for any graph  $H$  such that  $\chi(H) = k$ , we have  $\chi_k(G) = \chi(G \circ H)$ . The simplest choice of  $H$  is often  $K_k$ , the complete graph of order  $k$ .

For bounds on the chromatic number of (lexicographic) graph products, we refer readers to [110, 165]. By the lexicographic product, any bound on  $\chi(G)$  can also be transformed to a bound on  $\chi_k(G)$ . In particular:

$$\chi_k(G) = \chi(G \circ K_k) \geq \omega(G \circ K_k) = \omega(G)\omega(K_k) = k\omega(G). \quad (2.1)$$

Here we use that  $\omega(G \circ H) = \omega(G)\omega(H)$  for general graphs  $G$  and  $H$ . We also mention the following result

$$\alpha(G \circ H) = \alpha(G)\alpha(H). \quad (2.2)$$

Both (2.1) and (2.2) are proven by Geller and Stahl [110]. Let us also state the following known result:

$$\chi(G \circ H) \leq \chi(G)\chi(H). \quad (2.3)$$

Inequality (2.3) can be derived as follows. Denote the vertex sets of  $G$  and  $H$  by  $V(G)$  and  $V(H)$ , respectively. For an optimal coloring of  $G$  and  $H$ , define  $c(u)$  as the color of some vertex  $u$ . Graph  $G \circ H$  has vertices  $(g_i, h_i)$ . Every vertex in  $G \circ H$  can then be assigned a 2-color combination  $(c(g_i), c(h_i))$ . Note that, by interpreting these 2-color combinations as simply colors, this constitutes a coloring of  $G \circ H$  by using  $\chi(G)\chi(H)$  colors. Combining inequalities (2.1) and (2.3) results in

$$k\omega(G) \leq \chi_k(G) = \chi(G \circ K_k) \leq k\chi(G). \quad (2.4)$$

The inequalities in (2.4) may be strict. An example is the cycle graph with five vertices and  $k = 2$ , as  $\chi_2(C_5) = 5$  [288]. Note that, by (2.4), any upper bound on  $\chi(G)$  can be transformed into an upper bound on  $\chi_k(G)$ . To compute (or approximate)  $\chi_k(G)$  one can consult the wide range of existing literature on standard graph coloring by using  $\chi_k(G) = \chi(G \circ K_k)$ . Next to that, more specific literature on multicoloring can also be examined. Campêlo et al. [48] present an integer linear programming formulation for the  $k$ -multicoloring of a graph and study the facial structure of the corresponding polytope. Malaguti and Toth [214] use a combination of tabu search and population management procedures as a metaheuristic to solve (slightly generalized) multicoloring problems. Mehrotra and Trick [224] apply branch and price to generate stable sets for solving the multicoloring problem.

Narasimhan and Manber [232] generalize  $\vartheta(G)$  by introducing  $\vartheta_k(G)$ , which can be defined as follows. Define, for a graph  $G$  on  $n$  vertices, the set of matrices

$$\mathcal{A}(G) := \{A \in \mathcal{S}^n : A_{ii} = 1 \ \forall i \in [n], A_{ij} = 1 \ \forall \{i, j\} \notin E(G)\}. \quad (2.5)$$

For any  $X \in \mathcal{S}^n$ , we order its eigenvalues as  $\lambda_{\max}(X) = \lambda_1(X) \geq \dots \geq \lambda_n(X) = \lambda_{\min}(X)$ . Then,

$$\vartheta_k(G) := \min_{A \in \mathcal{A}(G)} \sum_{i=1}^k \lambda_i(A), \quad k \in [n]. \quad (2.6)$$

Narasimhan and Manber prove that

$$\alpha_k(\overline{G}) \leq \vartheta_k(\overline{G}) \leq \chi_k(G) \quad (2.7)$$

and thus also  $\omega_k(G) \leq \vartheta_k(\overline{G}) \leq \chi_k(G)$ . Recall that  $\alpha_k(G)$  is the cardinality of the largest subset  $C \subseteq V$  such that the subgraph induced in  $G$  by  $C$ , denoted  $G[C]$ , satisfies  $\chi(G[C]) \leq k$ . Inequality (2.7) generalizes the Lovász' sandwich theorem [206].

Alizadeh [7] derived the following SDP formulation of  $\vartheta_k(G)$ , see also Fan's theorem [83] and [172]:

$$\begin{aligned} \vartheta_k(G) = & \min_{\mu \in \mathbb{R}, X, Y \in \mathcal{S}^n} \langle \mathbf{I}, Y \rangle + \mu k \\ \text{s.t.} & X_{ij} = 0 \quad \forall \{i, j\} \notin E(G) \\ & \mu \mathbf{I} + X - \mathbf{J} + Y \succeq 0, Y \succeq 0. \end{aligned} \quad (\vartheta_k\text{-SDP})$$

The dual problem for  $\vartheta_k\text{-SDP}$  is:

$$\begin{aligned} \vartheta_k(G) = & \max_{Y \in \mathcal{S}^n} \langle \mathbf{J}, Y \rangle \\ \text{s.t.} & Y_{ij} = 0 \quad \forall \{i, j\} \in E(G) \\ & \langle \mathbf{I}, Y \rangle = k, 0 \preceq Y \preceq \mathbf{I}. \end{aligned} \quad (\vartheta_k\text{-SDP2})$$

Note that for  $k = 1$  constraint  $Y \preceq \mathbf{I}$  is redundant. We now show that  $\vartheta_k(\overline{G}) \leq \chi_k(G)$ , using arguments that are different from those used in [232]. In an optimal  $k$ -multicoloring of  $G$ , define for each of the  $\chi_k(G)$  colors used a vector  $\mathbf{y}^j \in \{0, 1, k\}^{n+1}$ ,  $1 \leq j \leq \chi_k(G)$ . For the entries of  $\mathbf{y}^j$ , we have  $\mathbf{y}_0^j = k$  and  $\mathbf{y}_i^j = 1$  if vertex  $i$  has color  $j$ , 0 otherwise. Then

$$\frac{1}{k^2} \sum_{j=1}^{\chi_k(G)} \mathbf{y}^j (\mathbf{y}^j)^\top = \begin{bmatrix} \chi_k(G) & \mathbf{1}_n^\top \\ \mathbf{1}_n & \frac{1}{k} \mathbf{I}_n + \frac{1}{\chi_k(G)} X \end{bmatrix},$$

for some  $X \in \mathcal{S}^n$  satisfying  $X_{ij} = 0$  for all  $\{i, j\} \in E(G)$ . By the Schur complement we find  $\frac{\chi_k(G)}{k} \mathbf{I} + X - \mathbf{J} \succeq 0$ . Simply set  $Y = \mathbf{0} \in \mathcal{S}^n$ . Then the triple  $(\frac{\chi_k(G)}{k}, X, Y)$  is feasible for  $\vartheta_k\text{-SDP}$  (for  $\overline{G}$ ) with objective value  $\chi_k(G)$ .

To conclude this section we state the following result:

$$\vartheta_k(\overline{G}) \leq k\vartheta(\overline{G}) \leq \chi_k(G). \quad (2.8)$$

Narasimhan and Manber [232] prove the first inequality in (2.8). To show this, let  $\tilde{A} \in \mathcal{A}(\overline{G})$  such that  $\lambda_1(\tilde{A}) = \vartheta(\overline{G})$ . Then  $\vartheta_k(\overline{G}) \leq \sum_{i=1}^k \lambda_i(\tilde{A}) \leq k\lambda_1(\tilde{A})$  and the proof follows. The second inequality in (2.8) follows from  $\vartheta(\overline{G \circ K_k}) = k\vartheta(\overline{G})$  and  $\vartheta(\overline{G \circ K_k}) \leq \chi(G \circ K_k) = \chi_k(G)$ . The second inequality in (2.8) also follows from the following known results  $\vartheta(\overline{G}) \leq \chi_f(G)$  and  $k\chi_f(G) \leq \chi_k(G)$  where  $\chi_f(G)$  is the fractional chromatic number of a graph, see e.g., [47]. In this chapter, we show that  $\vartheta_k(G) = k\vartheta(G)$  for many highly symmetric graphs.

## 2.3 The sequence $(\vartheta_k(G))_{k \in [n]}$

In this section we consider the sequence  $\vartheta_1(G), \vartheta_2(G), \dots, \vartheta_n(G)$  where  $G$  is a graph on  $n$  vertices. We first prove that this sequence is bounded from above (Proposition 2.7) and increasing (Proposition 2.8). Then, we prove that the increments of the sequence i.e.,  $\vartheta_k(G) - \vartheta_{k-1}(G)$  are decreasing in  $k$ , see Theorem 2.10. We also show that this increment can be arbitrarily small for a particular graph, see Theorem 2.11. Let us first establish a relation between  $\vartheta_k(G)$  and  $\chi(G)$ .

**Proposition 2.7.** *For  $G = (V, E)$  and  $k \geq \chi(G)$ , we have  $\vartheta_k(G) = |V|$ . Furthermore,  $\vartheta_k(G) \leq \min\{k\vartheta(G), |V|\}$  for all  $k \leq |V|$ .*

*Proof.* Let  $k \geq \chi(G)$ . Then  $\alpha_k(G) = |V|$ , where we take the  $k$  stable sets to be the color classes in an optimal coloring of  $G$ . Thus, it follows from (2.7) that  $|V| \leq \vartheta_k(G)$ .

Furthermore, note that for any graph  $G$ , matrix  $\mathbf{J}_{|V|} \in \mathcal{A}(G)$  is feasible for (2.6). Since matrix  $\mathbf{J}_{|V|}$  has eigenvalue  $|V|$  with multiplicity one and the other eigenvalues equal to 0, we have  $\vartheta_k(G) \leq |V|$  for any graph  $G$ . Therefore, when  $k \geq \chi(G)$  we have  $\vartheta_k(G) = |V|$ . Besides,  $\vartheta_k(G) \leq k\vartheta(G)$  by (2.8).  $\square$

Part of Proposition 2.7 can be more succinctly stated as  $\vartheta_{\chi(G)}(G) = |V|$ . Proposition 2.7 also shows that the sequence  $(\vartheta_k(G))_{k \leq |V|}$  is bounded from above by  $|V|$ . The next proposition shows that this sequence is non-decreasing in  $k$ .

**Proposition 2.8.** *For any graph  $G$  on  $n$  vertices,  $\vartheta_k(G) \leq \vartheta_{k+1}(G)$  for any integer  $k < n$ , with equality if and only if  $\vartheta_k(G) = n$ .*

*Proof.* Consider a graph  $G$  on  $n$  vertices and let  $Y$  be optimal for  $\vartheta_k$ -SDP2, with  $k < n$ . We have  $\text{tr}(Y) = k$  and  $0 \preceq Y \preceq \mathbf{I}_n$ . Define  $Z := \left(1 - \frac{1}{n-k}\right)Y + \frac{1}{n-k}\mathbf{I}_n$ . It follows that  $Z$  is feasible for  $\vartheta_{k+1}$ -SDP2 and thus

$$\vartheta_{k+1}(G) \geq \langle \mathbf{J}_n, Z \rangle = \vartheta_k(G) + \frac{n - \vartheta_k(G)}{n - k} \geq \vartheta_k(G),$$

with equality if and only if  $\vartheta_k(G) = n = |V(G)|$ .  $\square$

Proposition 2.8 allows us to further restrict  $\vartheta_k$ -SDP.

**Proposition 2.9.** *Let  $G$  be an arbitrary graph on  $n$  vertices, and  $(X^*, Y^*, \mu^*)$  be an optimal solution to  $\vartheta_k$ -SDP for  $k < n$ . Then  $\mu^* \geq 0$ . If  $k = n$ , then  $(X, Y, \mu) = (\mathbf{0}, \mathbf{J}, 0)$  is an optimal solution to  $\vartheta_k$ -SDP.*

*Proof.* For  $k < n$ , we prove the statement by contradiction. Assume that  $(X^*, Y^*, \mu^*)$  is optimal for  $\vartheta_k$ -SDP, with  $\mu^* < 0$ . Note that  $(X^*, Y^*, \mu^*)$  is also feasible for  $\vartheta_{k+1}$ -SDP. Since  $\mu^* < 0$ , this would imply that  $\vartheta_k(G) > \vartheta_{k+1}(G)$ , which contradicts Proposition 2.8. Thus  $\mu^* \geq 0$ . If  $k = n$ , then  $(X, Y, \mu) = (\mathbf{0}, \mathbf{J}, 0)$  is feasible for  $\vartheta_k$ -SDP and attains objective value  $n$ , which is optimal since  $\vartheta_k(G) = n$  by Proposition 2.7.  $\square$

Next, we investigate the increments of the sequence  $(\vartheta_k(G))_{k \in [n]}$ . For that purpose, we define for any graph  $G$  and  $k \geq 1$  the increment of  $(\vartheta_k(G))_{k \in [n]}$  as follows:

$$\Delta_k(G) := \vartheta_k(G) - \vartheta_{k-1}(G) \text{ if } k > 1, \text{ and } \Delta_1(G) := \vartheta_1(G) \quad (2.9)$$

**Theorem 2.10.** *For any graph  $G$  on  $n$  vertices and  $k \in [n - 1]$ ,  $\Delta_k(G) \geq \Delta_{k+1}(G)$ .*

*Proof.* For  $k \in [n]$ , where  $n$  is the number of vertices of  $G$ , let the matrices  $A_k \in \mathcal{A}(G)$ , see (2.5), satisfy

$$\sum_{i=1}^k \lambda_i(A_k) = \vartheta_k(G). \quad (2.10)$$

Stated differently,  $A_k$  is an optimal solution to (2.6) for computing  $\vartheta_k(G)$ . Since (2.6) is a minimization problem,

$$\vartheta_k(G) \leq \sum_{i=1}^k \lambda_i(A_{k'}), \quad k' \in [n]. \quad (2.11)$$

By substituting (2.10) and (2.11) in the definition of  $\Delta_k(G)$  for  $k \geq 2$ , see (2.9), we obtain:

$$\Delta_k(G) = \vartheta_k(G) - \vartheta_{k-1}(G) \leq \sum_{i=1}^k \lambda_i(A_{k-1}) - \sum_{i=1}^{k-1} \lambda_i(A_{k-1}) = \lambda_k(A_{k-1}). \quad (2.12)$$

Similarly,

$$\Delta_k(G) = \vartheta_k(G) - \vartheta_{k-1}(G) \geq \sum_{i=1}^k \lambda_i(A_k) - \sum_{i=1}^{k-1} \lambda_i(A_k) = \lambda_k(A_k). \quad (2.13)$$

Combining (2.12) and (2.13) yields  $\Delta_k(G) \geq \lambda_k(A_k) \geq \lambda_{k+1}(A_k) \geq \Delta_{k+1}(G)$ ,  $k \geq 2$ . The inequality  $\Delta_1(G) \geq \Delta_2(G)$  follows from (2.8).  $\square$

Let us summarize the implications of Proposition 2.8 and Theorem 2.10. Proposition 2.8 proves that

$$\Delta_k(G) = 0 \iff \vartheta_{k-1}(G) = |V|, \quad \forall k \in \{2, 3, \dots, |V|\}. \quad (2.14)$$

For complete graphs we have  $\Delta_k(K_n) = 1$ , see Theorem 2.15. There exist however graphs for which  $\Delta_k(G) < 1$ . We investigate the limiting behaviour of  $\Delta_k(G)$  in Section 2.3.1.

When we consider the sequence induced by  $\vartheta_k(G)$  as a function of  $k$ , we know that this sequence is increasing towards  $|V(G)|$ . Theorem 2.10 shows that the increments in this sequence decrease in  $k$ . Loosely speaking, one might say the second derivative of  $f(k) = \vartheta_k(G)$  is negative.

### 2.3.1 Limiting behaviour of $\Delta_k(G)$

In this section we show that, for any real number  $\varepsilon > 0$ , there exists a graph  $G$  and a number  $k \geq 1$  such that  $0 < \Delta_k(G) < \varepsilon$ . For this purpose, define graph  $\mathcal{G}_n = (V(\mathcal{G}_n), E(\mathcal{G}_n))$ ,  $n \in \mathbb{N}$ , as follows:

$$V(\mathcal{G}_n) := [n] \text{ and } E(\mathcal{G}_n) := \{\{i, j\} : 1 \leq i < j \leq n-1\} \cup \{\{n-1, n\}\}. \quad (2.15)$$

Graph  $\mathcal{G}_n$  is thus a complete graph on  $n-1$  vertices plus one additional vertex. This additional vertex is connected to the complete graph  $K_{n-1}$  by a single edge.

**Theorem 2.11.** For  $n \geq 5$ , we have  $\vartheta_{n-2}(\mathcal{G}_n) = n - 2 + \frac{2}{n-3}\sqrt{(n-2)(n-4)}$ .

*Proof.* We prove the result by finding a lower and upper bound on  $\vartheta_{n-2}(\mathcal{G}_n)$ , both of which equal  $n - 2 + \frac{2}{n-3}\sqrt{(n-2)(n-4)}$ . Let  $p := \sqrt{\frac{n-4}{(n-2)(n-3)^2}}$ . Define matrix  $Y \in \mathcal{S}^n$  as follows:

$$Y = \begin{bmatrix} \frac{n-4}{n-3}\mathbf{I}_{n-2} & \mathbf{0}_{n-2} & p\mathbf{1}_{n-2} \\ \mathbf{0}_{n-2}^\top & 1 & 0 \\ p\mathbf{1}_{n-2}^\top & 0 & \frac{1}{n-3} \end{bmatrix}.$$

Matrix  $Y$  is feasible for  $\vartheta_{n-2}$ -SDP2 (see Page 25) if  $0 \preceq Y \preceq \mathbf{I}$ . Therefore we derive

$$\mathbf{I} - Y = \begin{bmatrix} \frac{1}{n-3}\mathbf{I}_{n-2} & \mathbf{0}_{n-2} & -p\mathbf{1}_{n-2} \\ \mathbf{0}_{n-2}^\top & 0 & 0 \\ -p\mathbf{1}_{n-2}^\top & 0 & \frac{n-4}{n-3} \end{bmatrix},$$

and take the Schur complement of the block  $\frac{1}{n-3}\mathbf{I}_{n-2}$  of  $\mathbf{I} - Y$ :

$$\begin{bmatrix} 0 & 0 \\ 0 & \frac{n-4}{n-3} \end{bmatrix} - \begin{bmatrix} \mathbf{0}_{n-2}^\top \\ p\mathbf{1}_{n-2}^\top \end{bmatrix} (n-3)\mathbf{I}_{n-2} \begin{bmatrix} \mathbf{0}_{n-2} & p\mathbf{1}_{n-2} \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \succeq 0.$$

Thus  $Y \preceq \mathbf{I}$ . Similarly, by taking the Schur complement of the upper left  $(n-1) \times (n-1)$  block matrix of  $Y$ , we find that  $Y \succeq 0$ . We omit the details of this computation. This implies that  $Y$  is feasible for  $\vartheta_{n-2}$ -SDP2 and

$$\vartheta_{n-2}(\mathcal{G}_n) \geq \langle \mathbf{J}, Y \rangle = n - 2 + \frac{2}{n-3}\sqrt{(n-2)(n-4)}. \quad (2.16)$$

Finding the (equal) upper bound on  $\vartheta_{n-2}(\mathcal{G}_n)$  is a bit more involved. Set

$$A := \begin{bmatrix} \gamma\mathbf{J}_{n-2} + (1-\gamma)\mathbf{I}_{n-2} & \mathbf{0}_{n-2} & \mathbf{1}_{n-2} \\ \mathbf{0}_{n-2}^\top & 1 & 0 \\ \mathbf{1}_{n-2}^\top & 0 & 1 \end{bmatrix}, \text{ for } \gamma := \frac{n-5}{n-3}\sqrt{\frac{n-2}{n-4}}.$$

Note that  $A \in \mathcal{A}(\mathcal{G}_n)$ , see (2.5). We show that for

$$\begin{aligned} \beta_1 &:= \frac{-(n-3)\gamma + \sqrt{(n-3)^2\gamma^2 - 4(2-n)}}{2} = \sqrt{\frac{n-2}{n-4}}, \\ \beta_2 &:= \frac{-(n-3)\gamma - \sqrt{(n-3)^2\gamma^2 - 4(2-n)}}{2} = -\sqrt{(n-2)(n-4)}, \end{aligned} \quad (2.17)$$

the vectors  $v_i = [\mathbf{1}_{n-2}^\top, 0, \beta_i]^\top$ ,  $i \in \{1, 2\}$ , are two eigenvectors of matrix  $A$ . Consider

$$Av_i = \begin{bmatrix} ((n-3)\gamma + 1 + \beta_i)\mathbf{1}_{n-2} \\ 0 \\ \left(\frac{n-2}{\beta_i} + 1\right)\beta_i \end{bmatrix}, \quad i \in \{1, 2\}. \quad (2.18)$$

By (2.17) we have that  $\beta_i$ ,  $i \in \{1, 2\}$ , are the roots of the equation  $\beta^2 + (n-3)\gamma\beta + (2-n) = 0$ , and so  $\beta_i + (n-3)\gamma = (n-2)/\beta_i$ . Then, the right-hand side of (2.18) equals  $v_i$  scaled by the corresponding eigenvalue, which is given by:

$$(n-3)\gamma + 1 + \beta_i = \frac{n-2}{\beta_i} + 1.$$

In particular, the two corresponding eigenvalues are given by

$$\frac{n-2}{\beta_1} + 1 = \sqrt{(n-2)(n-4)} + 1 \text{ and } \frac{n-2}{\beta_2} + 1 = 1 - \sqrt{\frac{n-2}{n-4}}. \quad (2.19)$$

Also  $[\mathbf{0}_{n-2}^\top, 1, 0]^\top$  is an eigenvector of  $A$  with corresponding eigenvalue one (and multiplicity one). Since  $A$  is a real symmetric matrix, its eigenvectors are orthogonal. The remaining  $n-3$  eigenvectors are thus  $w_i = [c_i^\top, 0, 0]^\top$ ,  $i \in [n-3]$ , where  $c_i \in \mathbb{R}^{n-2}$  is a vector whose entries sum to 0. The eigenvectors  $w_i$  correspond to eigenvalues of  $1 - \gamma$ . We have provided all eigenvectors and eigenvalues of  $A$ , see also (2.19). The four unique eigenvalues of  $A$  are ordered as follows:

$$\sqrt{(n-2)(n-4)} + 1 > 1 > 1 - \gamma > 1 - \sqrt{\frac{n-2}{n-4}}, \quad (2.20)$$

with corresponding multiplicities 1, 1,  $n-3$ , 1, respectively. The sum of the largest  $n-2$  eigenvalues of  $A$  serves as upper bound on  $\vartheta_{n-2}(\mathcal{G}_n)$ , see (2.6). Therefore, it follows by (2.20) that

$$\begin{aligned} \vartheta_{n-2}(\mathcal{G}_n) &\leq \sum_{i=1}^{n-2} \lambda_i(A) = \sqrt{(n-2)(n-4)} + 2 + (n-4)\gamma \\ &= n-2 + \frac{2}{n-3} \sqrt{(n-2)(n-4)}. \end{aligned} \quad (2.21)$$

The upper bound on  $\vartheta_{n-2}(\mathcal{G}_n)$  given by (2.21) coincides with the lower bound (2.16), which proves the theorem.  $\square$

Using Theorem 2.11 we can show that  $\Delta_{n-1}(\mathcal{G}_n)$  ( $n \geq 5$ ) converges to zero. Indeed,

$$\Delta_{n-1}(\mathcal{G}_n) = \vartheta_{n-1}(\mathcal{G}_n) - \vartheta_{n-2}(\mathcal{G}_n) \leq 2 \left( 1 - \frac{1}{n-3} \sqrt{(n-2)(n-4)} \right), \quad (2.22)$$

where we have also used that  $\vartheta_{n-1}(\mathcal{G}_n) \leq n$ , as proven by Proposition 2.7. From (2.22) it follows that  $\Delta_{n-1}(\mathcal{G}_n)$  ( $n \geq 5$ ) converges to zero as  $n$  goes to infinity. To conclude, strictly positive values of  $\Delta_k(G)$  can be arbitrarily small. It is unclear whether lower bounds exist on  $\Delta_k(G)$  for fixed  $k$ . One example of such a bound is simple for  $k=1$  i.e.,  $\Delta_1(G) = \vartheta_1(G) \geq \alpha(G) \geq 1$ .

## 2.4 Graph products and the generalized $\vartheta$ -number

In this section we present bounds for  $\vartheta_k(G)$  when  $G$  is the strong product of two graphs (Theorem 2.13) and the disjunction product of two graphs (Theorem 2.14).

In [204], Lovász proved the following result:

$$\vartheta(G_1 \boxtimes G_2) = \vartheta(G_1)\vartheta(G_2), \quad (2.23)$$

where  $G_1 \boxtimes G_2$  is the strong product of  $G_1$  and  $G_2$ , see Definition 2.1. Since  $G_1 \boxtimes K_k$  is isomorphic to  $G_1 \circ K_k$  and  $\vartheta(K_k) = 1$  we have that

$$\vartheta(G \circ K_k) = \vartheta(G \boxtimes K_k) = \vartheta(G) \leq \vartheta_k(G),$$

where the inequality follows from Proposition 2.8 and the fact that  $\vartheta(G) = \vartheta_1(G)$ . Below, we generalize (2.23) to  $\vartheta_k$ . For that purpose we need the following well known result, see, e.g., [143].

**Lemma 2.12.** *For square matrices  $A$  and  $B$  with eigenvalues  $\lambda_i$  and  $\mu_j$  respectively, the eigenvalues of the Kronecker product  $A \otimes B$  equal  $\lambda_i\mu_j$ , and  $\text{tr}(A \otimes B) = \text{tr}(A)\text{tr}(B)$ .*

**Theorem 2.13.** *For any graphs  $G_1$  and  $G_2$*

$$\frac{1}{k}\vartheta_k(G_1)\vartheta_k(G_2) \leq \vartheta_k(G_1 \boxtimes G_2) \leq k\vartheta(G_1)\vartheta(G_2). \quad (2.24)$$

*Proof.* Let  $X_1^*$  and  $X_2^*$  be optimal to  $\vartheta_k$ -SDP2 for  $G_1$  and  $G_2$  respectively. The adjacency matrix of  $G_1 \boxtimes G_2$  is given by

$$A_{G_1 \boxtimes G_2} = (A_{G_1} + \mathbf{I}) \otimes (A_{G_2} + \mathbf{I}) - \mathbf{I},$$

see e.g., [269]. Consider  $Y = \frac{1}{k}X_1^* \otimes X_2^*$ . From the adjacency matrix of  $G_1 \boxtimes G_2$  it can be verified that  $Y_{ij} = 0, \forall \{i, j\} \in E(G_1 \boxtimes G_2)$ . By Lemma 2.12 and the fact that  $0 \preceq X_i^* \preceq \mathbf{I}, i \in \{1, 2\}$ , the eigenvalues of  $Y$  lie between 0 and 1 and thus  $0 \preceq Y \preceq \mathbf{I}$ . Additionally,  $\text{tr}(Y) = (1/k)\text{tr}(X_1^*)\text{tr}(X_2^*) = k$ . It follows that matrix  $Y$  is feasible for  $\vartheta_k$ -SDP2 for  $G_1 \boxtimes G_2$  and attains the following objective value:

$$\langle \mathbf{J}, Y \rangle = \frac{1}{k}\langle \mathbf{J}, X_1^* \otimes X_2^* \rangle = \frac{1}{k}\langle \mathbf{J}, X_1^* \rangle \langle \mathbf{J}, X_2^* \rangle = \frac{1}{k}\vartheta_k(G_1)\vartheta_k(G_2).$$

This proves the lower bound in (2.24). The upper bound in (2.24) follows from combining (2.8) and (2.23).  $\square$

The bounds from Theorem 2.13 are attained, for example, when both  $G_1$  and  $G_2$  are complete graphs (see Theorem 2.15). In general, the bounds for  $\vartheta_k(G_1 \boxtimes G_2)$  from Theorem 2.13 are more loose for larger values of  $k$ .

We now focus on the disjunction graph product (see Definition 2.1). For graphs  $G_1$  and  $G_2$  of order  $n_1$  and  $n_2$  respectively, we consider the adjacency matrix  $A_{G_1 \vee G_2} \in \{0, 1\}^{n_1 n_2 \times n_1 n_2}$ . We partition  $A_{G_1 \vee G_2}$  in  $n_1^2$  square submatrices  $A^{u_1, v_1} \in \{0, 1\}^{n_2 \times n_2}$ , for  $u_1, v_1 \in V(G_1)$ . If  $\{u_1, v_1\} \in E(G_1)$ , then  $A^{u_1, v_1} = \mathbf{J}_{n_2}$  by the definition of  $\vee$ . If  $\{u_1, v_1\} \notin E(G_1)$ , then  $A^{u_1, v_1} = A_{G_2}$ . It follows that

$$A_{G_1 \vee G_2} = \min \{A_{G_1} \otimes \mathbf{J}_{n_2} + \mathbf{J}_{n_1} \otimes A_{G_2}, \mathbf{J}_{n_1 n_2}\},$$

where the minimum is defined to be entrywise. Our next result provides an upper bound on the generalized  $\vartheta$ -number for the disjunction product of two graphs.

**Theorem 2.14.** *For graphs  $G_1$  and  $G_2$  of orders  $n_1$  and  $n_2$  respectively, we have*

$$\vartheta_k(G_1 \vee G_2) \leq \min \{n_1\vartheta_k(G_2), n_2\vartheta_k(G_1)\}.$$

*Proof.* Consider the SDP problem  $\vartheta_k$ -SDP2 for  $G_1 \vee G_2$ . This maximization problem is least constrained when  $G_1 = \overline{K}_{n_1}$ . Thus

$$\vartheta_k(G_1 \vee G_2) \leq \vartheta_k(\overline{K}_{n_1} \vee G_2). \quad (2.25)$$

We will show that  $\vartheta_k(\overline{K}_{n_1} \vee G_2) = n_1\vartheta_k(G_2)$ . Let  $X^*$  be an optimal solution to  $\vartheta_k$ -SDP2 for  $G_2$ . Matrix  $\mathbf{J}_{n_1} \otimes \frac{1}{n_1}X^*$  is a feasible solution to  $\vartheta_k$ -SDP2 for  $\overline{K}_{n_1} \vee G_2$ . The objective value of this solution equals

$$\begin{aligned} \langle \mathbf{J}, \mathbf{J}_{n_1} \otimes \frac{1}{n_1}X^* \rangle &= \frac{1}{n_1} \langle \mathbf{J}_{n_1} \otimes \mathbf{J}_{n_1}, \mathbf{J}_{n_2} \otimes X^* \rangle = \frac{1}{n_1} \langle \mathbf{J}_{n_1}, \mathbf{J}_{n_1} \rangle \langle \mathbf{J}_{n_2}, X^* \rangle \\ &= n_1\vartheta_k(G_2) \implies \vartheta_k(\overline{K}_{n_1} \vee G_2) \geq n_1\vartheta_k(G_2). \end{aligned} \quad (2.26)$$

Let  $(Y^*, X^*, \mu^*)$  be an optimal solution to  $\vartheta_k$ -SDP for  $G_2$  with  $\mu^* \geq 0$ , see Proposition 2.9. Then  $\mathbf{J}_{n_1} \otimes Y^*$ ,  $\mathbf{J}_{n_1} \otimes X^*$  and  $n_1\mu^*$  form a feasible solution to  $\vartheta_k$ -SDP for  $\overline{K}_{n_1} \vee G_2$ . Namely, by Lemma 2.12 we have that  $\mathbf{J}_{n_1} \otimes Y^* \succeq 0$ . Also

$$\begin{aligned} n_1\mu^*\mathbf{I} + \mathbf{J}_{n_1} \otimes X^* - \mathbf{J} + \mathbf{J}_{n_1} \otimes Y^* &= \\ \mu^*(n_1\mathbf{I}_{n_1} - \mathbf{J}_{n_1}) \otimes \mathbf{I}_{n_2} + \mathbf{J}_{n_1} \otimes (\mu^*\mathbf{I}_{n_2} + X^* - \mathbf{J}_{n_2} + Y^*) &\succeq 0, \end{aligned}$$

where we have used

that  $\mu^* \geq 0$ . Lastly, this feasible solution to the minimization problem obtains an objective value of

$$\begin{aligned} \langle \mathbf{I}, \mathbf{J}_{n_1} \otimes Y^* \rangle + n_1\mu^*k &= n_1 (\langle \mathbf{I}, Y^* \rangle + \mu^*k) = n_1\vartheta_k(G_2) \\ \implies \vartheta_k(\overline{K}_{n_1} \vee G_2) &\leq n_1\vartheta_k(G_2). \end{aligned} \quad (2.27)$$

Now (2.26) and (2.27) imply that  $\vartheta_k(\overline{K}_{n_1} \vee G_2) = n_1\vartheta_k(G_2)$ . This equality, combined with (2.25), proves that

$$\vartheta_k(G_1 \vee G_2) \leq n_1\vartheta_k(G_2). \quad (2.28)$$

From the definition of the disjunction graph product (see Definition 2.1), it follows that the disjunction graph product is commutative and thus

$$\vartheta_k(G_1 \vee G_2) = \vartheta_k(G_2 \vee G_1) \leq n_2\vartheta_k(G_1). \quad (2.29)$$

Combining equations (2.28) and (2.29) proves the theorem.  $\square$

The proof shows that when either  $G_1$  or  $G_2$  is the complement of a complete graph, graph  $G_1 \vee G_2$  attains the bound of Theorem 2.14.

## 2.5 Value of $\vartheta_k$ for some graphs

In [204], Lovász derived an explicit expression for the  $\vartheta$ -number of cycle graphs and the Kneser graphs. In this section, we derive the generalized  $\vartheta$ -number for those graphs, as well as for circulant, complete, complete multipartite graphs, and the Johnson graphs. In Section 2.5.1 we present bounds for  $\vartheta_k(G)$  when  $G$  is a regular graph and show that the bound is tight for edge-transitive graphs. Section 2.5.2 provides an analysis of  $\vartheta(K_k \square G)$ , which is an upper bound on the number of vertices in the maximum  $k$ -colorable subgraph of  $G$ .

**Theorem 2.15.** *For  $k \leq n$ ,  $\vartheta_k(K_n) = k$ .*

*Proof.* Consider the SDP problem  $\vartheta_k$ -SDP2. For the complete graph, the only matrices feasible for  $\vartheta_k$ -SDP2 are diagonal matrices with trace equal to  $k$ . Set for instance  $Y = \frac{k}{n}\mathbf{I}$ . Then  $Y$  is feasible for  $\vartheta_k$ -SDP2 and has objective value  $k$ .  $\square$

For cycle graphs it is known, see [288, Thm. 6], that  $\chi_k(C_{2n+1}) = 2k + 1 + \lfloor \frac{k-1}{n} \rfloor$ , and  $\chi_k(C_{2n}) = 2k$ . The latter equality admits a simple proof, see e.g. (2.77), unlike the first. Since  $C_n$  is bipartite when  $n$  is even, it follows from Proposition 2.7 that  $\vartheta_k(C_n) = n$  for all  $k \in \{2, 3, \dots, n\}$ . To compute  $\vartheta_2(C_n)$  for odd cycle graphs, we require the following lemma.

**Lemma 2.16.** *For odd  $n \geq 5$ , we have that*

$$0.447 \approx \frac{\sqrt{5}}{5} \leq \frac{\vartheta(C_n)}{n} < \frac{\vartheta(C_{n+2})}{n+2} < \frac{1}{2}.$$

*Proof.* By [204, Corollary 5] we have, for odd  $n$ , that

$$\frac{\vartheta(C_n)}{n} = f(n), \text{ where } f(x) := \frac{\cos(\pi/x)}{1 + \cos(\pi/x)}.$$

For  $x > 1$ , the derivative of  $f$  is given by

$$f'(x) = \frac{\pi \sin(\frac{\pi}{x})}{(1 + \cos(\frac{\pi}{x}))^2 x^2}, \quad x > 1,$$

and so  $f$  is increasing on  $(1, \infty)$ . In particular,  $f$  is increasing on  $[5, \infty)$ . Therefore, for  $n$  an odd integer in  $[5, \infty)$ , we have that

$$\frac{\sqrt{5}}{5} = f(5) \leq f(n) < f(n+2) < \lim_{x \rightarrow \infty} f(x) = \frac{1}{2},$$

which proves the result.  $\square$

Let us introduce a circulant matrix and an edge-transitive graph. We need both terms in the proof of the following theorem. Each row of a circulant matrix equals the preceding row in the matrix rotated one element to the right. Circulant matrices thus have a constant row sum. This constant row sum is also one of the eigenvalues with  $\mathbf{1}$  as its corresponding eigenvector. A graph is edge transitive if its automorphism group acts transitively on edges, i.e., if for every two edges there is an automorphism that maps one to the other.

**Theorem 2.17.** *Let  $n$  be odd and  $n > 1$ . Then  $\vartheta_2(C_n) = 2\vartheta(C_n)$  and  $\vartheta_k(C_n) = n$  for all  $k \geq 3$ .*

*Proof.* For  $n = 3$ ,  $C_3 = K_3$  and the result follows from Theorem 2.15. Thus let  $n \geq 5$ . Let  $\Gamma \subset \mathcal{S}^n$  be the set of optimal feasible solutions to  $\vartheta_1$ -SDP2 for  $C_n$  and let  $Y \in \Gamma$ . Note  $\Gamma$  is convex. Let  $p(Y)$  denote an optimal solution to  $\vartheta_1$ -SDP2 obtained by permuting the vertices of  $C_n$  by automorphism  $p$ . Matrix  $p(Y) \in \Gamma$ . Denote the average over all automorphisms  $p$  by  $\bar{Y}$ . Then  $\bar{Y} \in \Gamma$  by convexity of  $\Gamma$  and since  $C_n$  is edge transitive,  $\bar{Y}$  is a circulant matrix, like the adjacency matrix of  $C_n$ .

As  $\bar{Y} \in \Gamma$ , we find

$$\langle \mathbf{J}, \bar{Y} \rangle = \text{tr}(\mathbf{1}\mathbf{1}^\top \bar{Y}) = \text{tr}(\mathbf{1}^\top \bar{Y} \mathbf{1}) = \mathbf{1}^\top \bar{Y} \mathbf{1} = \vartheta_1(C_n). \quad (2.30)$$

As  $\bar{Y}$  is also circulant, it has eigenvector  $\mathbf{1}$ . By (2.30), its corresponding eigenvalue equals  $\bar{\lambda} = \vartheta(C_n)/n$ .

We will prove that the largest eigenvalue of  $\bar{Y}$  equals  $\bar{\lambda}$ . Assume that the largest eigenvalue of  $\bar{Y}$  does not equal  $\bar{\lambda}$ . Then  $\bar{Y}$  has eigenvalue  $\tilde{\lambda}$ , for some  $\tilde{\lambda} > \bar{\lambda}$ . Since  $\bar{Y}$  is a symmetric circulant matrix of odd dimension,  $\bar{Y}$  has only one eigenvalue with odd multiplicity [294]. Thus  $\tilde{\lambda}$  or  $\bar{\lambda}$  have multiplicity greater than one. Note that since  $\bar{Y}$  is feasible for  $\vartheta_1$ -SDP2, it has nonnegative eigenvalues that sum to one. However, both terms  $\tilde{\lambda} + 2\bar{\lambda}$  and  $2\tilde{\lambda} + \bar{\lambda}$  are strictly greater than one by Lemma 2.16, and hence, the assumption that  $\bar{\lambda}$  is not the largest eigenvalue of  $\bar{Y}$  leads to a contradiction.

The largest eigenvalue of  $\bar{Y}$  is thus smaller than  $1/2$ . Then  $2\bar{Y} \preceq \mathbf{I}$ . Clearly,  $2\bar{Y}$  satisfies the other feasibility conditions of  $\vartheta_2$ -SDP2. Thus  $2\bar{Y}$  is feasible for  $\vartheta_2$ -SDP2 and  $\vartheta_2(C_n) \geq 2\vartheta(C_n)$ . Combined with (2.8), it follows that  $\vartheta_2(C_n) = 2\vartheta(C_n)$ .

The proof of  $\vartheta_3(C_n) = n$  follows from combining  $\chi(C_n) = 3$  and Proposition 2.7.  $\square$

Graphs for which the adjacency matrix is a circulant matrix are called *circulant graphs*, like the cycle graphs and some Paley graphs. There has been research done on computing  $\vartheta(G)$  for circulant graphs [21, 39, 40, 62]. In particular, Crespi [62] computes the Lovász theta number for the circulant graphs of degree four having even displacement, while Brimkov et al. [40] consider  $\vartheta(C_{n,j})$ , where  $V(C_{n,j}) = \{0, 1, \dots, n-1\}$  and  $E(C_{n,j}) = E(C_n) \cup \{\{i, i'\} : i - i' \equiv j \pmod{n}\}$ .

Let  $H_n$  be a connected circulant graph on  $n$  vertices. Then  $H_n$  contains a Hamiltonian cycle [31]. Equivalently, the cycle graph  $C_n$  is a minor of  $H_n$ . Maximization problem  $\vartheta_k$ -SDP2 is then more restricted for  $H_n$  than it is for  $C_n$ . Thus

$$\vartheta_1(H_n) \leq \vartheta_1(C_n) \leq \frac{n}{2}.$$

Consider  $\vartheta_1$ -SDP2 for  $H_n$ . Graph  $H_n$  has a circulant adjacency matrix, meaning we can restrict optimization of  $\vartheta_1$ -SDP2 over the Lee scheme, the association scheme of symmetric circulant matrices, without loss of generality [108]. As (2.30) shows,  $\vartheta_1$ -SDP2 is now equivalent to maximizing the largest (scaled) eigenvalue over feasible matrices. Let  $M$  be a matrix optimal for  $\vartheta_1$ -SDP2 for graph  $H_n$ . Then  $\lambda_{\max}(M) = \vartheta(H_n)/n \leq 1/2$ . Then  $2M$  is also optimal for  $\vartheta_2$ -SDP2 for graph  $H_n$ . More generally, if  $k \leq n/\vartheta(H_n)$ , then  $\lambda_{\max}(kM) \leq 1$  and  $kM$  is then feasible for  $\vartheta_k$ -SDP2, attaining

the objective value  $\min\{k\vartheta(H_n), n\}$ . In case  $k > n/\vartheta(H_n)$ , we have  $\vartheta_k(H_n) = n$ . Thus, in general

$$\vartheta_k(H_n) = \min\{k\vartheta(H_n), n\}. \quad (2.31)$$

For any  $k$ , there exists a circulant graph  $P$  on  $n$  vertices such that  $\vartheta_k(P) < n$ . Specifically, if  $P$  is the Paley graph on  $n$  vertices, then  $\vartheta(P) = \sqrt{n}$  (cf. [129]). For fixed  $k$  and  $n$  large enough,  $k\sqrt{n} < n$ .

**Theorem 2.18.** *For  $m_1 \geq m_2 \geq \dots \geq m_p$  and  $k \leq p$ ,*

$$\vartheta_k(K_{m_1, \dots, m_p}) = \sum_{i=1}^k m_i.$$

*Proof.* Let us write  $K = K_{m_1, \dots, m_p}$ , with corresponding adjacency matrix  $A_K$ . Note that  $A_K = \mathbf{J}_n - \text{Diag}(\mathbf{J}_{m_1}, \dots, \mathbf{J}_{m_p})$ , and that  $X := \text{Diag}(\mathbf{J}_{m_1}, \dots, \mathbf{J}_{m_p}) \in \mathcal{A}(K)$ , see (2.5). Therefore  $X$  is feasible for (2.6). The eigenvalues of  $X$  are the eigenvalues of the block matrices  $\mathbf{J}_{m_i}$ . Then,  $\lambda_i(X) = m_i$  for  $i \in [p]$ . Thus, we have that

$$\vartheta_k(K) \leq \sum_{i=1}^k \lambda_i(X) = \sum_{i=1}^k m_i. \quad (2.32)$$

Note that  $\alpha_k(K) = \sum_{i=1}^k m_i$ . Then the proof follows from combining (2.7) and (2.32).  $\square$

Recall again the definition of  $\Delta_k(G)$ , given in (2.9). We have shown in Section 2.3.1 that strictly positive values  $\Delta_k(G)$  can be arbitrarily small. We show now, by use of Theorem 2.18, that the ratio between strictly positive successive values of  $\Delta_k(G)$  can be arbitrarily small. More formally, for any  $\varepsilon > 0$  and any  $k \geq 1$ , there exists a graph  $G$  such that

$$0 < \frac{\Delta_{k+1}(G)}{\Delta_k(G)} < \varepsilon. \quad (2.33)$$

We again ignore the case  $\Delta_k(G) = 0$ , see (2.14). In view of Theorem 2.18, we have

$$\frac{\Delta_2(K_{n,1})}{\Delta_1(K_{n,1})} = \frac{1}{n} < \varepsilon,$$

for some integer  $n$  sufficiently large. Thus for sufficiently large  $n$ , graph  $K_{n,1}$  satisfies (2.33) for  $k = 1$ . Graph  $K_{n,n,1}$  satisfies (2.33) for  $k = 2$ . Graph  $K_{n,n,n,1}$  satisfies (2.33) for  $k = 3$ , and so on.

### 2.5.1 Regular graphs

In this section we present an upper bound on the  $\vartheta_k$ -number for regular graphs, see Theorem 2.20. This result can be seen as a generalization of the Lovász upper bound on the  $\vartheta$ -number for regular graphs. We exploit the result of Theorem 2.20 to prove that  $\vartheta_k(G) = k\vartheta(G)$  when  $G$  is the Johnson graph, see Theorem 2.22. Moreover, we

derive an explicit expression for the generalized theta number for the Kneser graphs, see Corollary 2.24.

Let us first state the following result.

**Theorem 2.19** ([204]). *For a regular graph  $G$  on  $n$  vertices, having adjacency matrix  $A_G$  and eigenvalues  $\lambda_1(A_G) \geq \lambda_2(A_G) \geq \dots \geq \lambda_n(A_G)$ , we have that*

$$\vartheta(G) \leq \frac{n\lambda_n(A_G)}{\lambda_n(A_G) - \lambda_1(A_G)}. \quad (2.34)$$

*Inequality (2.34) holds with equality if  $G$  is also edge-transitive.*

For a finite set of real numbers  $P$ , we denote by  $\mathbf{S}_k(P)$  the sum of the largest  $k$  elements in  $P$ . For any symmetric matrix  $X$ , we denote by  $\Lambda(X)$  the eigenspectrum of  $X$ , i.e., the set of (possibly repeated) eigenvalues of  $X$ . The following idea involving  $\mathbf{S}_k(\cdot)$  and  $\Lambda(\cdot)$  is important for the rest of this section: if  $G$  is a regular graph on  $n$  vertices, its adjacency matrix  $A_G$  has eigenvector  $\mathbf{1}_n$ , corresponding to the largest eigenvalue. Since  $A_G$  is symmetric, its other eigenvectors are orthogonal to  $\mathbf{1}_n$ . Thus, the eigenvectors of  $A_G$  and  $\mathbf{J}_n$  coincide. Therefore, if  $G$  is a regular graph with adjacency matrix  $A_G$  and  $\lambda_i = \lambda_i(A_G)$ ,  $i \in [n]$ , we have that

$$\Lambda(\mathbf{J}_n + xA_G) = \{n + x\lambda_1, x\lambda_2, \dots, x\lambda_n\}, \lambda_1 \geq \dots \geq \lambda_n, x \in \mathbb{R}. \quad (2.35)$$

For  $k \in [n]$  and  $x \in \mathbb{R}$ , we define the continuous function

$$g_k(x) := \mathbf{S}_k(\Lambda(\mathbf{J} + xA_G)) = \mathbf{S}_k(\{n + x\lambda_1, x\lambda_2, \dots, x\lambda_n\}), x \in \mathbb{R}. \quad (2.36)$$

Let us provide some simple results on  $g_k(x)$ . The function  $g_k$  is convex (Lemma B.1 on Page 204). Note that  $g_1(x) = \max\{n + x\lambda_1, x\lambda_2, \dots, x\lambda_n\}$  and  $g_n(x) = n + x \sum_{i=1}^n \lambda_i = n + x \operatorname{tr}(A_G) = n$ . For  $k \in \{2, 3, \dots, n-1\}$  and  $x \leq 0$ , it can be observed that  $g_k(x) = x \sum_{i=n-k+2}^n \lambda_i + \max\{n + x\lambda_1, x\lambda_{n-k+1}\}$ . We have that  $n + x\lambda_1 = x\lambda_{n-k+1}$  if and only if  $x = \frac{n}{\lambda_{n-k+1} - \lambda_1}$ . Therefore, for  $k \in \{2, 3, \dots, n-1\}$ ,

$$g_k(x) = \begin{cases} x \sum_{i=n-k+1}^n \lambda_i & \text{if } x < \frac{n}{\lambda_{n-k+1} - \lambda_1}, \\ n + x \left( \lambda_1 + \sum_{i=n-k+2}^n \lambda_i \right) & \text{if } \frac{n}{\lambda_{n-k+1} - \lambda_1} \leq x \leq 0, \\ n + x \sum_{i=1}^k \lambda_i & \text{if } x > 0. \end{cases} \quad (2.37)$$

Now, we state our result, which we prove using the function  $g_k$  as in (2.36).

**Theorem 2.20.** *For any regular graph  $G$  on  $n$  vertices and  $k \in [n]$ , we have that*

$$\vartheta_k(G) \leq \min \left\{ n + \frac{n(\lambda_1(A_G) + \sum_{i=n-k+2}^n \lambda_i(A_G))}{\lambda_{n-k+1}(A_G) - \lambda_1(A_G)}, n \right\}, \quad (2.38)$$

*where we set the summation equal to 0 when  $k = 1$ . Inequality (2.38) holds with equality if  $G$  is also edge-transitive.*

*Proof.* The proof is an extension of Lovász' [204] proof of Theorem 2.19. Note that Theorem 2.19 is equivalent to (2.38) for  $k = 1$ . Indeed, for  $k = 1$ , (2.34) and (2.38) are equivalent. For  $k = n$ , we have that  $\vartheta_k(G) = n$  by Proposition 2.7. It remains to prove the result for  $k \in \{2, 3, \dots, n - 1\}$ .

Let  $G$  be a regular graph on  $n$  vertices and  $k \in \{2, 3, \dots, n - 1\}$ . Note that  $\mathbf{J}_n + xA_G \in \mathcal{A}(G)$ , see (2.5), for any  $x \in \mathbb{R}$ . Therefore, it follows from the definition of  $\vartheta_k$ , see (2.6), that  $\vartheta_k(G) \leq \min_{x \in \mathbb{R}} g_k(x)$ , for  $g_k$  as in (2.36). To minimize  $g_k$ , we use (2.37) and the fact that  $g_k$  is convex (see Lemma B.1 on Page 204).

Let  $\lambda_1 \geq \dots \geq \lambda_n$  be the eigenvalues of  $A_G$ . Since the  $\lambda_i$  sum to  $\text{tr}(A_G) = 0$ , it follows that the term  $\sum_{i=n-k+1}^n \lambda_i$  in (2.37) is negative. It can similarly be shown that the term  $\sum_{i=1}^k \lambda_i$  in (2.37) is positive. Thus,  $\lim_{x \rightarrow -\infty} g_k(x) = \lim_{x \rightarrow \infty} g_k(x) = \infty$ . By the convexity of  $g_k$ , its minimum must then be attained at one of the two breakpoints of  $g_k$ , that is either  $x = \frac{n}{\lambda_{n-k+1} - \lambda_1}$  or  $x = 0$ . Hence,

$$\vartheta_k(G) \leq \min_{x \in \mathbb{R}} g_k(x) = \min \left\{ g_k \left( \frac{n}{\lambda_{n-k+1} - \lambda_1} \right), g_k(0) \right\}. \quad (2.39)$$

Inequality (2.39) is equivalent to (2.38).

We now prove that (2.38) holds with equality when  $G$  is edge-transitive. It is known that the sum of the  $k$  largest eigenvalues of a matrix, as in (2.6), is a convex function [244]. Thus the average over all optimal solutions to (2.6) of all automorphisms of  $G$  is also optimal. Since  $G$  is edge-transitive, this average is of the form  $\mathbf{J} + xA_G$ , which proves the equality claim.  $\square$

We remark that Theorem 2.17 can also be proven by applying Theorem 2.20.

**Corollary 2.21.** *Let  $G$  be a regular, edge-transitive graph on  $n$  vertices, and let  $k \in [n]$ . If the smallest eigenvalue of its adjacency matrix has multiplicity  $k$  or greater, then  $\vartheta_k(G) = \min\{k\vartheta(G), n\}$ .*

*Proof.* Let us write  $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n$  for the eigenvalues of  $A_G$ . Since the multiplicity of  $\lambda_n$  is at least  $k$ , it follows that

$$\sum_{i=n-k+2}^n \lambda_i = (k-1)\lambda_n \text{ and } \lambda_{n-k+1} = \lambda_n. \quad (2.40)$$

As  $G$  is regular and edge-transitive, (2.38) holds with equality. Substituting (2.40) in (2.38) yields that  $\vartheta_k(G) = \min\{\theta, n\}$ , where

$$\begin{aligned} \theta &:= n + \frac{n(\lambda_1 + \sum_{i=n-k+2}^n \lambda_i)}{\lambda_{n-k+1} - \lambda_1} = n + \frac{n(\lambda_1 + (k-1)\lambda_n)}{\lambda_n - \lambda_1} \\ &= n + \frac{n\lambda_1}{\lambda_n - \lambda_1} + (k-1)\vartheta(G) = \frac{n\lambda_n}{\lambda_n - \lambda_1} + (k-1)\vartheta(G) = k\vartheta(G). \end{aligned} \quad (2.41)$$

For the third and fourth equalities in (2.41), we have used that  $\vartheta(G) = n\lambda_n/(\lambda_n - \lambda_1)$  by Theorem 2.19.  $\square$

We apply Corollary 2.21 to the Johnson graphs (Definition 2.4). Note that Johnson graphs are edge-transitive (see e.g., [53]) and regular.

**Theorem 2.22.** *Let integers  $n, m$  satisfy  $1 \leq m \leq \frac{n}{2}$  and let  $f \in \{0, 1, \dots, m\}$ . For the Johnson graph  $J(n, m, f)$ , we have that*

$$\vartheta_k(J(n, m, f)) = \min \left\{ k\vartheta(J(n, m, f)), \binom{n}{m} \right\}, \quad k \in [n-1]. \quad (2.42)$$

*Proof.* Let  $A$  be the adjacency matrix of  $J(n, m, f)$ . Note that that  $n$  here does not refer to the number of vertices of  $J(n, m, f)$ , but to a parameter of the graph. The number of vertices is given by  $\binom{n}{m}$ .

Since the Johnson graphs are edge-transitive and regular, we may apply Corollary 2.21. Hence, (2.42) follows by showing that the smallest eigenvalue of  $A$  has multiplicity  $n-1$  or greater. It is known that the multiplicities of the eigenvalues of  $A$  are given by 1 (corresponding to the largest eigenvalue) and  $\binom{n}{j} - \binom{n}{j-1}$  for  $j \in [m]$ , cf. e.g. [43] or [73]. The multiplicity of the smallest eigenvalue is unknown for general  $n, m$  and  $f$ . It is clear however, that this multiplicity is at least

$$\mu := \min_{j \in [m]} \mu_j, \quad \text{for } \mu_j := \binom{n}{j} - \binom{n}{j-1}.$$

We continue the proof by considering two cases based on the value of  $n$ .

**Case 1.**  $n \geq 2$  and  $n \neq 4$ .

If  $n \neq 4$ , then  $\mu = \mu_1 = n-1$  [272, Thm. 1]. Since  $k \leq \mu$ , (2.42) follows from Corollary 2.21.

**Case 2.**  $n = 4$ .

We check the finite number of Johnson graphs  $J(4, m, f)$ , with  $1 \leq m \leq n/2 = 2$  and  $f \in \{0, 1, \dots, m\}$ . Note that for any Johnson graph, the eigenvalues of its adjacency matrix, and their corresponding multiplicities, have closed form expressions, see e.g., [44]. Thus, the  $\vartheta$ -number of Johnson graphs can also be expressed in closed form, using Theorem 2.19.

1.  $J(4, 1, 0) = \overline{J(4, 1, 1)} = K_4$ .
2.  $J := J(4, 2, 0)$  is a bipartite graph, which implies that  $\chi(J) = 2$ , and so  $\vartheta_2(J) = \binom{4}{2} = 6$  by Proposition 2.7. Moreover,  $\vartheta_1(J) = \vartheta(J) = 3$ . Lastly,  $\vartheta_3(J) = \binom{4}{2} = 6$  by Proposition 2.8.
3.  $J := J(4, 2, 1)$ . The smallest eigenvalue of  $A$  has multiplicity 2. Then Corollary 2.21 proves that  $\vartheta_2(J) = 2\vartheta(J) = 4$ . Since  $\chi(J) = 3$ ,  $\vartheta_3(J) = 6$ .
4.  $J(4, 2, 2) = \overline{K_6}$ . □

The  $\vartheta$ -number of Johnson graphs has also been studied in [199]. When the Johnson graph parameters  $n, m$  and  $f$  satisfy certain conditions, the smallest eigenvalue of its adjacency matrix can be expressed in closed form. In that case, we can slightly strengthen Theorem 2.22.

**Lemma 2.23.** *Let integers  $n, m$  and  $f \in \{0, 1, \dots, m-1\}$  satisfy*

$$1 \leq m \leq \frac{n}{2} \text{ and } 0 \leq f \leq \frac{m(m-1)}{n-1}. \quad (2.43)$$

For the Johnson graph  $J(n, m, f)$ , we have that

$$\vartheta_k(J(n, m, f)) = \min \left\{ k \binom{n}{m} \frac{fn - m^2}{fn - nm}, \binom{n}{m} \right\}, \quad k \leq \binom{n}{m}. \quad (2.44)$$

*Proof.* We write  $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_v$  for the eigenvalues of the adjacency matrix of  $J := J(n, m, f)$ , where  $v := \binom{n}{m}$  equals the number of vertices of  $J$ . Since  $J$  is a regular graph,  $\lambda_1$  equals the degree of any of its vertices. When  $n, m$  and  $f$  satisfy the conditions of the lemma, see (2.43), a closed form expression for  $\lambda_v$  is known [44, Thm. 3.10], and its multiplicity equals  $n-1$  [73]. In particular,

$$\lambda_1 = \binom{m}{f} \binom{n-m}{m-f} > 0, \quad \lambda_v = \lambda_1 \frac{fn - m^2}{m(n-m)} < 0. \quad (2.45)$$

Substituting the expressions for  $\lambda_1$  and  $\lambda_v$  in (2.34) yields that

$$\vartheta(J) = \binom{n}{m} \frac{fn - m^2}{fn - nm}. \quad (2.46)$$

Note that  $n$  in (2.34) refers to the number of vertices, which equals  $v = \binom{n}{m}$  for graph  $J$ . Substituting (2.46) in (2.42) proves (2.44) for the case  $k \leq n-1$ .

Let now  $k = n$ . Theorem 2.20 states that  $\vartheta_k(J) = \min\{\theta, \binom{n}{m}\}$ , where

$$\theta := v + \frac{v(\lambda_1 + \sum_{i=v-n+2}^v \lambda_i)}{\lambda_{v-n+1} - \lambda_1}. \quad (2.47)$$

We consider the term  $\lambda_1 + \sum_{i=v-n+2}^v \lambda_i$  in (2.47). Since the multiplicity of  $\lambda_v$  equals  $n-1$ , it follows that  $\sum_{i=v-n+2}^v \lambda_i = (n-1)\lambda_v$ . Hence,

$$\lambda_1 + \sum_{i=v-n+2}^v \lambda_i = \lambda_1 + (n-1)\lambda_v = \lambda_1 \left( 1 + (n-1) \frac{fn - m^2}{m(n-m)} \right). \quad (2.48)$$

We have used (2.45) for the second equality in (2.48). By using the condition  $f \leq \frac{m(m-1)}{n-1}$ , we find that

$$1 + (n-1) \frac{fn - m^2}{m(n-m)} \leq 0 \implies \lambda_1 + \sum_{i=v-n+2}^v \lambda_i \leq 0 \implies \theta \geq v = \binom{n}{m}.$$

Thus,  $\vartheta_n(J) = \min\{\theta, \binom{n}{m}\} = \binom{n}{m}$ . Then also  $\vartheta_k(J) = \binom{n}{m}$  for all  $k \geq n$  by Proposition 2.8.  $\square$

Since the Kneser graph  $K(n, m)$  is isomorphic to  $J(n, m, 0)$ , we can easily extend Lemma 2.23 to Kneser graphs.

**Corollary 2.24.** *Let  $1 \leq m \leq n/2$  for integers  $m$  and  $n$ . For the Kneser graph  $K(n, m)$  and  $k \leq \binom{n}{m}$ , we have that*

$$\vartheta_k(K(n, m)) = \min \left\{ k \binom{n-1}{m-1}, \binom{n}{m} \right\}. \quad (2.49)$$

*Proof.* We have that  $K(n, m) = J(n, m, f)$ , for  $f = 0$ . Then the parameters  $n$ ,  $m$  and  $f$  satisfy the conditions of Lemma 2.23, see (2.43). Note also that for  $f = 0$ , it holds that  $\binom{n}{m} \frac{fn-m^2}{fn-nm} = \binom{n}{m} \frac{m}{n} = \binom{n-1}{m-1}$ . Thus, expressions (2.42) and (2.49) coincide for  $f = 0$ .  $\square$

### 2.5.2 Relation between $\vartheta(K_k \square G)$ and $\vartheta_k(G)$

Gvozdenović and Laurent [130] show how to exploit an upper bound on the independence number of a graph to obtain a lower bound on the chromatic number of its complement graph. They do not consider the generalized  $\vartheta$ -number in the bounding procedure. Kuryatnikova et al. [172] exploit the generalized  $\vartheta$ -number to compute bounds on the chromatic number of a graph. For some graphs, the lower bounds on  $\chi(G)$  from [172] coincide with the bounds obtained by using the theta number as suggested by [130]. Here we explain that finding by analyzing  $\vartheta(K_k \square G)$  for symmetric graphs. We also show that the gap between  $\vartheta_k(G)$  and  $\vartheta(K_k \square G)$  can be arbitrarily large.

Chvátal [58] noted that

$$\alpha_k(G) = |V(G)| \iff \chi(G) \leq k.$$

Stated differently,  $\chi(G) = \min \{k : k \in \mathbb{N}, \alpha_k(G) = |V(G)|\}$ , or in words, the  $k$  stable sets giving  $\alpha_k(G)$  correspond to the color classes of  $G$  in an optimal coloring. Analogue to  $\chi_k(G) = \chi(G \circ K_k)$ , it is known (cf. [172]) that

$$\alpha_k(G) = \alpha(K_k \square G), \quad (2.50)$$

where  $K_k \square G$  is the Cartesian graph product, see Definition 2.1. For a graph parameter  $\beta(G)$  that satisfies

$$\alpha(G) \leq \beta(G) \leq \chi(\overline{G}),$$

Gvozdenović and Laurent [130] define  $\Psi_\beta(G)$  as follows:

$$\Psi_\beta(G) := \min \{k : k \in \mathbb{N}, \beta(K_k \square G) = |V(G)|\}.$$

Then  $\Psi_\alpha(G) = \chi(G)$ . The operator  $\Psi_\beta(\cdot)$  can be applied to a variety of graph parameters  $\beta(G)$  and enables obtaining a hierarchy of bounds for  $\chi(G)$  from a hierarchy of bounds for  $\alpha(G)$ . For example, when  $\beta(G) = \vartheta(G)$  Gvozdenović and Laurent [130] show that  $\Psi_\vartheta(G) = \lceil \vartheta(\overline{G}) \rceil$ . It follows from (2.50) that parameters  $\vartheta_k(G)$  and  $\vartheta(K_k \square G)$  both provide upper bounds on  $\alpha_k(G)$ . Therefore, it is natural to compare  $\Psi_\vartheta(G)$  with the similar parameter

$$\Phi_{\vartheta_k}(G) := \min \{k : k \in \mathbb{N}, \vartheta_k(G) = |V(G)|\}.$$

This comparison boils down to the comparison of  $\vartheta(K_k \square G)$  and  $\vartheta_k(G)$ . Numerical results in [172] suggest the following conjecture.

**Conjecture 2.25.** For any graph  $G$  and any natural number  $k$ ,  $\vartheta(K_k \square G) \leq \vartheta_k(G)$ .

We have numerically verified Conjecture 2.25 for all graphs on at most 7 vertices, and all valid values of  $k$ .

We show in Proposition 2.27 that the gap between  $\vartheta(K_k \square G)$  and  $\vartheta_k(G)$  can be arbitrarily large. We first state the following lemma that is needed in the rest of this section.

**Lemma 2.26** ([130]). *Given  $A, B \in \mathcal{S}^n$  and  $Y = \mathbf{I}_k \otimes A + (\mathbf{J}_k - \mathbf{I}_k) \otimes B$ , then  $Y \succeq 0$  if and only if  $A - B \succeq 0$  and  $A + (k - 1)B \succeq 0$ . Furthermore,  $\Lambda(Y) = \Lambda(A + (k - 1)B) \cup \Lambda(A - B)^{\{k-1\}}$ .*

Now, we are ready to present our result.

**Proposition 2.27.** *For any real number  $M$ , there exists a graph  $G$  and integer  $k$  such that*

$$\vartheta_k(G) - \vartheta(K_k \square G) \geq M.$$

*Proof.* Consider again graph  $\mathcal{G}_n$ , as defined in (2.15) for even  $n$  and set  $k = n/2$ . We will show that  $\vartheta_{n/2}(\mathcal{G}_n) - \vartheta(K_{n/2} \square \mathcal{G}_n)$  is increasing in  $n$ . Let  $p = 1/(2\sqrt{n-2})$  and consider first

$$X = \begin{bmatrix} \frac{1}{2}\mathbf{I}_{n-2} & \mathbf{0}_{n-2} & p\mathbf{1}_{n-2} \\ \mathbf{0}_{n-2}^\top & \frac{1}{2} & 0 \\ p\mathbf{1}_{n-2}^\top & 0 & \frac{1}{2} \end{bmatrix}.$$

Taking the Schur complement of the bottom right  $2 \times 2$  block of  $X$  shows that  $0 \preceq X \preceq \mathbf{I}$  (see the proof of Theorem 2.11 for more details). Combined with the fact that  $\langle \mathbf{I}, X \rangle = k$ , it follows that  $X$  is feasible for  $\vartheta_k$ -SDP2. Hence,

$$\vartheta_{n/2}(\mathcal{G}_n) \geq \langle \mathbf{J}, X \rangle = \frac{n}{2} + 2p(n-2) = \frac{n}{2} + \sqrt{n-2}. \quad (2.51)$$

As for  $\vartheta(K_{n/2} \square G)$ , let

$$A = \begin{bmatrix} -k\mathbf{J}_{n-1} + (k+1)\mathbf{I}_{n-1} & \mathbf{1}_{n-1} \\ \mathbf{1}_{n-1}^\top & 1 \end{bmatrix}, \quad B = \begin{bmatrix} \mathbf{J}_{n-1} & \mathbf{1}_{n-1} \\ \mathbf{1}_{n-1}^\top & -k \end{bmatrix},$$

and set  $Y := \mathbf{I} \otimes A + (\mathbf{J} - \mathbf{I}) \otimes B$ . Then matrix  $Y \in \mathcal{A}(K_{n/2} \square G)$ , see (2.5). Furthermore, matrix  $Y$  is of the form described in Lemma 2.26. Then the largest eigenvalue of  $Y$  satisfies  $\lambda_1(Y) = \max\{\lambda_1(A - B), \lambda_1(A + (k - 1)B)\}$ . Similar to the methods used in the proof of Theorem 2.11, it can be shown that  $\lambda_1(Y) = k + 1$ . Thus,

$$\vartheta(K_{n/2} \square \mathcal{G}_n) \leq \lambda_1(Y) = \frac{n}{2} + 1. \quad (2.52)$$

Combining (2.51) and (2.52) for fixed  $M$  and large enough (even)  $n$ , gives  $\vartheta_{n/2}(\mathcal{G}_n) - \vartheta(K_{n/2} \square \mathcal{G}_n) \geq \sqrt{n-2} - 1 \geq M$ .  $\square$

We prove Conjecture 2.25 only for a particular class of graphs. Let us first show the following result.

**Theorem 2.28.** *Let  $G$  be graph on  $n$  vertices that is both edge-transitive and vertex-transitive. Then*

$$\vartheta(K_k \square G) = \min\{k\vartheta(G), n\} = \vartheta_k(G).$$

*Proof.* For notational convenience we denote  $A = A_G$ . Because  $G$  is regular, edge-transitive and vertex-transitive, we may assume without loss of generality that the set  $\mathcal{A}(K_k \square G)$ , see (2.5), contains only matrices of the form  $X = \mathbf{I}_k \otimes (\mathbf{J}_n + xA) + (\mathbf{J}_k - \mathbf{I}_k) \otimes (\mathbf{J}_n + y\mathbf{I}_n)$ . In order to minimize the largest eigenvalue of  $X$  we apply Lemma 2.26 and find

$$\begin{aligned} \lambda_1(X) = f(x, y) &= \max\{\lambda_1(xA - y\mathbf{I}_n), \lambda_1(k\mathbf{J}_n + xA + (k-1)y\mathbf{I}_n)\} \\ &= \max \begin{cases} f_1 = x\lambda_1 - y, \\ f_2 = x\lambda_n - y, \\ f_3 = kn + x\lambda_1 + (k-1)y, \\ f_4 = x\lambda_n + (k-1)y, \end{cases} \end{aligned}$$

where  $\lambda_1$  and  $\lambda_n$  are the greatest and smallest eigenvalue of  $A$  respectively. We have used that  $\Lambda(k\mathbf{J}_n + xA + (k-1)y\mathbf{I}_n)$  can be expressed similarly to (2.35). We minimize  $\lambda_1(X)$  by considering different intervals of  $x$ . In case  $x \geq 0$ , we have  $f(x, y) = \max\{f_1, f_3\}$ , which is minimized when  $x = 0$  and  $f_1 = f_3$ . Solving  $f_1 = f_3$  for  $y$ , when  $x = 0$ , yields  $y = -n$ . Thus, when  $x \geq 0$ , we find that  $f(0, -n) = n$  is the minimum. Furthermore,

$$\frac{kn}{\lambda_n - \lambda_1} \leq x \leq 0 \implies f(x, y) = \max\{f_2, f_3\}.$$

The minimum here is attained when

$$f_2 = f_3 \implies y = x \left( \frac{\lambda_n - \lambda_1}{k} \right) - n \implies f_2 = n + \frac{1}{k} ((k-1)\lambda_n + \lambda_1) x.$$

Depending on the sign of  $(k-1)\lambda_n + \lambda_1$  we find either  $f(0, -n) = n$  or  $f(\frac{kn}{\lambda_n - \lambda_1}, 0) = k\frac{n\lambda_n}{\lambda_n - \lambda_1} = k\vartheta(G)$ , by Theorem 2.19. For the case  $x \leq \frac{kn}{\lambda_n - \lambda_1}$ , note that

$$x \leq \frac{kn}{\lambda_n - \lambda_1} \implies f(x, y) = \max\{f_2, f_4\},$$

which is minimized when  $x = kn/(\lambda_n - \lambda_1)$  and  $f_2 = f_4$ . Solving  $f_2 = f_4$  for  $y$ , when  $x = kn/(\lambda_n - \lambda_1)$ , yields  $y = 0$  and thus  $f(\frac{kn}{\lambda_n - \lambda_1}, 0) = k\vartheta(G)$ . The minimum value of  $\lambda_1(X)$ , equivalently, the value  $\vartheta(K_k \square G)$ , thus equals  $\min\{k\vartheta(G), n\}$ .

Lastly, by edge-transitivity and vertex-transitivity of  $G$ , matrices optimal to  $\vartheta_k$ -SDP2 for  $G$  have a constant row sum. Thus, as (2.30) shows,  $\vartheta_k$ -SDP2 is then equivalent to maximizing the largest (scaled) eigenvalue over feasible matrices. Hence,  $\vartheta_k(G) = \min\{k\vartheta(G), n\}$ , as can be shown by derivations similar to those used for (2.31).  $\square$

A graph that is both edge-transitive and vertex-transitive is also known as a symmetric graph. Many Johnson graphs (Definition 2.4) are symmetric.

Kuryatnikova et al. [172] compute  $\vartheta_k(G)$  for several highly symmetric graphs (Table 13 in the online supplement to [172]). They remark that for those graphs,  $\Phi_{\vartheta_k}(G) = \lceil \vartheta(\overline{G}) \rceil$ . We explain this result for all the graphs present in Table 13 except for the graph  $H(12, 2, \{i : 1 \leq i \leq 7\})$  (see Section 2.8.1 for the notation). All the other graphs evaluated by Kuryatnikova et al. in Table 13 satisfy the assumptions of Theorem 2.28, hence,  $\vartheta_k(G) = \vartheta(K_k \square G)$  for those graphs. Therefore,

$$\Phi_{\vartheta_k}(G) = \Psi_{\vartheta}(G) = \lceil \vartheta(\overline{G}) \rceil. \quad (2.53)$$

Note that the Johnson graph  $J(n, m, m-1)$  is regular, vertex-transitive and edge-transitive. Therefore, equation (2.53) holds and  $\lceil \vartheta(J(n, m, m-1)) \rceil = n - m + 1$ .

## 2.6 Strongly regular graphs

In the previous section we showed that certain classes of graphs allow an analytical computation of  $\vartheta_k(G)$ . This section expands on the considered classes with strongly regular graphs, see Definition 2.6. We also derive an analogue of Theorem 2.28 for strongly regular graphs and the generalized  $\vartheta'$ -number, see Theorem 2.32.

Let  $G$  be an  $\text{srg}(n, d, \lambda, \mu)$ , and  $A_G$  its adjacency matrix. Since  $G$  is regular with valency  $d$ , we have that  $d$  is an eigenvalue of  $A_G$  with eigenvector  $\mathbf{1}$ . The matrix  $A_G$  has exactly two distinct eigenvalues associated with eigenvectors orthogonal to  $\mathbf{1}$ . These two eigenvalues are known as restricted eigenvalues and are usually denoted by  $r$  and  $s$ , where  $r \geq 0$  and  $s \leq -1$ . We consider here connected, non-complete, strongly regular graphs. For those graphs we have that  $s < -1$ . Thus, we exclude trivial cases.

Strongly regular graphs attain Lovász' bound of Theorem 2.19, see e.g., [131]. In particular, for a strongly regular graph  $G$  we have

$$\vartheta(G) = \frac{n\lambda_n(A_G)}{\lambda_n(A_G) - \lambda_1(A_G)}.$$

The following theorem provides an explicit expression for  $\vartheta_k(G)$  for strongly regular graphs.

**Theorem 2.29.** *For an  $\text{srg}(n, d, \lambda, \mu)$   $G$  with restricted eigenvalues  $r \geq 0$  and  $s < -1$ , we have*

$$\vartheta_k(G) = \min\{k\vartheta(G), n\} = \min\left\{k \frac{n\lambda_n(A_G)}{\lambda_n(A_G) - \lambda_1(A_G)}, n\right\}.$$

*Proof.* We prove the result by showing that the lower and upper bound on  $\vartheta_k(G)$  coincide. Consider  $\vartheta_k$ -SDP2, and set  $Y = \frac{k}{n}\mathbf{I} + xA_{\overline{G}}$ . When  $0 \preceq Y \preceq \mathbf{I}$ , the matrix  $Y$  is feasible for  $\vartheta_k$ -SDP2. These PSD constraints on  $Y$  can be rewritten in terms of  $x$ . As  $\vartheta_k$ -SDP2 is a maximization problem we may assume w.l.o.g.  $x \geq 0$ . Thus, for all  $i \leq n$ ,

$$\lambda_i(Y) = k/n + x\lambda_i(A_{\overline{G}}). \quad (2.54)$$

Since  $A_G$  has eigenvalues  $d \geq r > s$ , it follows that

$$\Lambda(A_{\overline{G}}) = \Lambda(\mathbf{J} - A_G - \mathbf{I}) = \{n - d - 1, -s - 1, -r - 1\}. \quad (2.55)$$

Substituting (2.55) in (2.54) and exploiting the fact that  $n-d-1 > -(s+1) > -(1+r)$  we have:

$$0 \preceq Y \preceq \mathbf{I} \iff 0 \leq \lambda_i(Y) \leq 1 \iff \begin{cases} k/n + x(-1-r) \geq 0 \\ k/n + x(n-d-1) \leq 1. \end{cases} \quad (2.56)$$

The last two inequalities in (2.56) provide upper bounds on  $x$ , i.e.,

$$x \leq \min \left\{ \frac{k}{n(1+r)}, \frac{n-k}{n(n-d-1)} \right\}. \quad (2.57)$$

When  $x$  satisfies (2.57),  $Y$  is thus feasible for  $\vartheta_k$ -SDP2 and  $\langle \mathbf{J}, Y \rangle$  will provide a lower bound on  $\vartheta_k(G)$ . In particular, with (2.57) at equality,

$$\langle \mathbf{J}, Y \rangle = k + n(n-d-1)x = \min \left\{ k \left( \frac{r+n-d}{1+r} \right), n \right\}. \quad (2.58)$$

Equation (2.58) implies that

$$\vartheta_k(G) \geq \min \left\{ k \left( \frac{r+n-d}{1+r} \right), n \right\}.$$

By (2.8) and Proposition 2.7, we have  $\vartheta_k(G) \leq \min\{k\vartheta(G), n\}$ . It remains only to show that

$$k \left( \frac{r+n-d}{1+r} \right) = k\vartheta(G). \quad (2.59)$$

The eigenvalues of  $A_G$  can be written in terms of the parameters of  $G$ , i.e.,

$$rs = \mu - d, \quad r + s = \lambda - \mu. \quad (2.60)$$

Furthermore, the parameters of any strongly regular graph satisfy

$$(n-d-1)\mu = d(d-\lambda-1), \quad (2.61)$$

see e.g., [42, Thm. 9.1.3]. Let us now rewrite the term:

$$\begin{aligned} \frac{r+n-d}{1+r} &= \frac{ns}{s-d} \frac{(r+n-d)(s-d)}{ns(1+r)} \\ &= \frac{ns}{s-d} \frac{ns + nrs + [d^2 - nd - (n-1)sr - d(r+s)]}{ns + nrs}. \end{aligned} \quad (2.62)$$

We evaluate the expression between the square brackets in (2.62) by using (2.60) and (2.61), which yields

$$\begin{aligned} &d^2 - nd - (n-1)rs - d(r+s) \\ &= d\lambda + d + (n-d-1)\mu - nd - (n-1)(\mu-d) - d(\lambda-\mu) = 0. \end{aligned}$$

Thus (2.62) equals  $ns/(s-d) = \vartheta(G)$ , which proves the theorem.  $\square$

Recall that in Section 2.5.2 we consider symmetric graphs (graphs that are both edge-transitive and vertex-transitive). Although there exist graphs that are both symmetric and strongly regular, note that neither one set of graphs is a subset of the other. The cycle graph  $C_6$  is an example of a graph that is symmetric, but not strongly regular. The strongly regular Chang graphs [52] provide an example of a strongly regular graph which is not symmetric.

In Section 2.5.2 we have proved that  $\vartheta(K_k \square G) = \min\{k\vartheta(G), n\} = \vartheta_k(G)$  holds for symmetric graphs, see Theorem 2.28. We show below that a similar relation holds for strongly regular graphs. In fact we prove a result for the generalized  $\vartheta'$ -number, denoted by  $\vartheta'_k(G)$ , that is the optimal value of the SDP relaxation  $\vartheta_k$ -SDP2 strengthened by adding nonnegativity constraints on the matrix variable. The generalized  $\vartheta'$ -number for  $k = 1$  is also known as the Schrijver's number [271].

To prove our result, we first present an SDP relaxation that relates  $\vartheta_k(G)$  and  $\vartheta'(K_k \square G)$ . Kuryatnikova et al. [172] introduce the following SDP relaxation

$$\begin{aligned} \theta_k^3(G) = & \max_{Y \in \mathcal{S}^n} \langle \mathbf{I}, Y \rangle \\ \text{s.t. } & Y_{ij} = 0 \quad \forall \{i, j\} \in E(G) \\ & Y_{ii} \leq 1 \quad \forall i \in [n] \\ & \begin{bmatrix} k & \text{diag}(Y)^\top \\ \text{diag}(Y) & Y \end{bmatrix} \succeq 0, \quad Y \geq 0, \end{aligned} \quad (2.63)$$

that provides an upper bound on  $\alpha_k(G)$ , the optimal value for the MkCS problem. The relaxation (2.63) can be simplified when  $G$  is a highly symmetric graph. In particular, if  $G$  is a strongly regular graph one can restrict optimization of the SDP relaxation (2.63) to feasible points in the coherent algebra spanned by  $\{\mathbf{I}, A, \mathbf{J} - \mathbf{I} - A\}$ . By applying symmetry reduction, (2.63) reduces to the following convex optimization problem:

$$\theta_k^3(G) := \max \quad ny_1 \quad (2.64a)$$

$$\text{s.t. } \quad y_1 + (n - d - 1)y_2 - \frac{n}{k}y_1^2 \geq 0 \quad (2.64b)$$

$$y_1 - (r + 1)y_2 \geq 0 \quad (2.64c)$$

$$y_1 - (s + 1)y_2 \geq 0 \quad (2.64d)$$

$$y_1 \leq 1 \quad (2.64e)$$

$$y_1, y_2 \geq 0. \quad (2.64f)$$

For details on symmetry reduction see e.g., [108, 172] and references therein.

In [172] the authors conjecture that  $\theta_k^3(G) \leq \vartheta'_k(G)$  for any graph  $G$ . Here we show that  $\theta_k^3(G) = \vartheta'_k(G)$  for any (non-trivial) strongly regular graph  $G$ .

**Lemma 2.30.** *Let  $G$  be an  $\text{srg}(n, d, \lambda, \mu)$  with restricted eigenvalues  $r \geq 0$  and  $s < -1$ . Then*

$$\theta_k^3(G) = \min \left\{ k \left( \frac{r + n - d}{r + 1} \right), n \right\} = \vartheta_k(G) = \vartheta'_k(G).$$

*Proof.* Note that for  $s < -1$  constraint (2.64d) is trivially satisfied. Points in which constraints (2.64b) and (2.64c) intersect are  $(0, 0)$  and  $\left( \frac{k(r+n-d)}{n(r+1)}, \frac{k(n+r-d)}{n(r+1)^2} \right)$ . The

first equality follows by combining the latter point and constraint (2.64e). The second equality follows from  $\vartheta(G) = (r + n - d)/(1 + r)$ , see (2.59), and Theorem 2.29. The third equality follows from (2.57) and the fact that  $\frac{k}{n(1+r)} \geq 0$  and  $\frac{n-k}{n(n-d-1)} \geq 0$ .  $\square$

It is known that  $\vartheta'(K_k \square G) \leq \theta_k^3(G)$ , see [172, Section 5.1]. We show below that equality holds when  $k < n(r+1)/(r+n-d)$  and  $G$  is a (non-trivial) strongly regular graph by proving an equivalence between the SDP relaxations that give  $\vartheta'(K_k \square G)$  and  $\theta_k^3(G)$ . The SDP relaxation for  $\vartheta'(K_k \square G)$ , see also  $\vartheta_k$ -SDP2, is invariant under permutations of  $k$  colors when the graph under the consideration is  $K_k \square G$ . This was exploited in [172] to derive the following symmetry reduced relaxation:

$$\begin{aligned} \vartheta'(K_k \square G) = & \max_{X, Z \in \mathcal{S}^n} \langle \mathbf{I}, X \rangle \\ \text{s.t.} & \quad X_{ij} = 0 \quad \forall \{i, j\} \in E(G) \\ & \quad Z_{ii} = 0 \quad \forall i \in [n] \\ & \quad X \geq 0, Z \geq 0, X - Z \succeq 0 \\ & \quad \begin{bmatrix} 1 & \text{diag}(X)^\top \\ \text{diag}(X) & X + (k-1)Z \end{bmatrix} \succeq 0. \end{aligned} \tag{2.65}$$

Relaxation (2.65) can be further simplified when  $G$  is a strongly regular graph. One can restrict optimization to the corresponding coherent algebra. By applying symmetry reduction, (2.65) reduces to the following optimization problem:

$$\begin{aligned} \vartheta'(K_k \square G) := & \max \quad nx_1 \\ \text{s.t.} & \quad x_1 + (n-d-1)x_2 - (dz_1 + (n-d-1)z_2) \geq 0 & (2.66a) \\ & \quad x_1 - (r+1)x_2 - (rz_1 - (r+1)z_2) \geq 0 & (2.66b) \\ & \quad x_1 - (s+1)x_2 - (sz_1 - (s+1)z_2) \geq 0 & (2.66c) \\ & \quad x_1 + (n-d-1)x_2 + \\ & \quad \quad (k-1)(dz_1 + (n-d-1)z_2) - nx_1^2 \geq 0 & (2.66d) \\ & \quad x_1 - (r+1)x_2 + (k-1)(rz_1 - (r+1)z_2) \geq 0 & (2.66e) \\ & \quad x_1 - (s+1)x_2 + (k-1)(sz_1 - (s+1)z_2) \geq 0 & (2.66f) \\ & \quad x_1 \leq 1 \\ & \quad x_1, x_2, z_1, z_2 \geq 0. \end{aligned}$$

Our next result relates optimization problems (2.64) and (2.66).

**Proposition 2.31.** *Let  $G$  be an  $\text{srg}(n, d, \lambda, \mu)$  with restricted eigenvalues  $r \geq 0$  and  $s < -1$ , and  $k < \frac{n(r+1)}{r+n-d}$ . Then the optimization problems (2.64) and (2.66) are equivalent.*

*Proof.* Let  $(x_1, x_2, z_1, z_2)$  be feasible for (2.66). We show that  $(y_1, y_2)$  where  $y_1 := x_1$  and  $y_2 := x_2$  is feasible for (2.64).

From (2.66a) and (2.66d) we have

$$\begin{cases} x_1 + (n-d-1)x_2 \geq (dz_1 + (n-d-1)z_2) \\ x_1 + (n-d-1)x_2 \geq nx_1^2 - (k-1)(dz_1 + (n-d-1)z_2), \end{cases}$$

from where it follows

$$x_1 + (n - d - 1)x_2 - \frac{n}{k}x_1^2 \geq \max \left\{ (dz_1 + (n - d - 1)z_2) - \frac{n}{k}x_1^2, (k - 1) \left( \frac{n}{k}x_1^2 - (dz_1 + (n - d - 1)z_2) \right) \right\}.$$

To verify that the lower bound above is nonnegative, note that either  $dz_1 + (n - d - 1)z_2 \geq \frac{n}{k}x_1^2$  or  $dz_1 + (n - d - 1)z_2 < \frac{n}{k}x_1^2$ . Therefore  $x_1 + (n - d - 1)x_2 - \frac{n}{k}x_1^2 \geq 0$  and constraint (2.64b) is satisfied.

Similarly, from (2.66b) and (2.66e) it follows that constraint (2.64c) is satisfied. Constraint (2.64d) is trivially satisfied by (2.66c) and (2.66f).

Conversely, let  $(y_1, y_2)$  be feasible for (2.64). Define  $x_1 := y_1$  and  $x_2 := y_2$ . Let  $z_1$  and  $z_2$  be the solutions of the following system of equations:

$$rz_1 = (r + 1)z_2, \quad dz_1 + (n - d - 1)z_2 = \frac{n}{k}x_1^2.$$

Thus,  $z_1 = \frac{n(r+1)}{k(d+r(n-1))}x_1^2$ ,  $z_2 = z_1 \frac{r}{r+1}$ . Therefore, constraint (2.66a) follows from (2.64b) and the construction of  $z_1$  and  $z_2$ . Similar arguments applied to (2.64c) can be used to verify that (2.66b) and (2.66e) are satisfied. To verify (2.66d) we rewrite the constraint as follows

$$\begin{aligned} & x_1 + (n - d - 1)x_2 + (k - 1)(dz_1 + (n - d - 1)z_2) - nx_1^2 = \\ & x_1 + (n - d - 1)x_2 - \frac{n}{k}x_1^2 + (k - 1) \left( dz_1 + (n - d - 1)z_2 - \frac{n}{k}x_1^2 \right) \geq 0. \end{aligned}$$

To verify constraint (2.66c) we exploit the construction of  $z_1$  and  $z_2$  as well as  $r \geq 0$  and  $s < -1$  to obtain:  $-(sz_1 - (s + 1)z_2) = \frac{r-s}{r+1}z_1 \geq 0$ . It remains to show that constraint (2.66f) is redundant for  $k < \frac{n(r+1)}{r+n-d}$ . Let us rewrite the constraint as follows

$$\begin{aligned} & x_1 - (s + 1)x_2 + (k - 1)(sz_1 - (s + 1)z_2) \\ & = x_1 - (s + 1)x_2 - \frac{n(k - 1)(r - s)}{k(d + r(n - 1))}x_1^2 \geq 0. \end{aligned} \tag{2.67}$$

A point of intersection of  $x_1 - (s + 1)x_2 - \frac{n(k-1)(r-s)}{k(d+r(n-1))}x_1^2 = 0$  and  $x_1 = (r + 1)x_2$  is  $\left( \frac{k(d+r(n-1))}{n(k-1)(r+1)}, \frac{k(d+r(n-1))}{n(k-1)(r+1)^2} \right)$ , and a point of intersection of  $x_1 + (n - d - 1)x_2 - \frac{n}{k}x_1^2 = 0$  and  $x_1 = (r + 1)x_2$  is  $\left( \frac{k(r+n-d)}{n(r+1)}, \frac{k(r+n-d)}{n(r+1)^2} \right)$ . Furthermore, an intersection point of  $x_1 + (n - d - 1)x_2 - \frac{n}{k}x_1^2 = 0$  and the  $x_1$ -axis is  $(\frac{k}{n}, 0)$ , and a point of intersection of  $x_1 - (s + 1)x_2 - \frac{n(k-1)(r-s)}{k(d+r(n-1))}x_1^2 = 0$  and the  $x_1$ -axis is  $\left( \frac{k(d+r(n-1))}{n(k-1)(r-s)}, 0 \right)$ . Note that the common intersection point of both parabolas,  $x_1 = (r + 1)x_2$  and the  $x_1$ -axis is  $(0, 0)$ . Let us find  $k$  for which

$$\frac{k(r + n - d)}{n(r + 1)} < \frac{k(d + r(n - 1))}{n(k - 1)(r + 1)} \iff k < \frac{n(r + 1)}{r + n - d}$$

and

$$\frac{k}{n} < \frac{k(d + r(n - 1))}{n(k - 1)(r - s)} \iff k < \frac{d + rn - s}{r - s}.$$

By using (2.62), one can verify that  $\frac{n(r+1)}{r+n-d} < \frac{d+rn-s}{r-s}$ , from where it follows that the constraint (2.67) is redundant when  $\frac{k(r+n-d)}{n(r+1)} < 1$ . It follows trivially that the objective values coincide for feasible solutions of two models that are related as described.  $\square$

The next result follows from the previous discussion.

**Theorem 2.32.** *Let  $G$  be an  $\text{srg}(n, d, \lambda, \mu)$  with restricted eigenvalues  $r \geq 0$  and  $s < -1$ , and  $k < \frac{n(r+1)}{r+n-d}$ . We have that  $\vartheta'(K_k \square G) = \vartheta'_k(G)$ .*

*Proof.* The proof follows from Lemma 2.30 and Proposition 2.31.  $\square$

## 2.7 Orthogonality graphs

In this section we compute the generalized  $\vartheta$ -number for the orthogonality graphs.

We motivate the study of orthogonality graphs by a scenario taken from [105]. Let  $n = 2^r$  for some  $r \geq 1$ . Consider a game where two players, Alice and Bob, each receive an  $n$ -dimensional binary vector as input. These two vectors are either equal or their Hamming distance (see Definition 2.2) equals  $n/2$ , that is, they differ in exactly  $2^{r-1}$  positions. Given these inputs, Alice and Bob must each return a  $r$ -dimensional binary vector as output. To win the game, Alice and Bob must return equal outputs if and only if their inputs were equal. Alice and Bob are not permitted to communicate once they receive their inputs. The players are, however, allowed to coordinate a strategy beforehand. One such strategy results in the definition of an orthogonality graph.

Vertices of the orthogonality graph  $\Omega_n$  are represented by the unique  $n$ -dimensional binary vectors. Vertices (equivalently vectors) are adjacent if their Hamming distance equals  $n/2$  and thus,  $\Omega_n = H(n, 2, \{n/2\})$ . Here  $H(n, 2, \{n/2\})$  denotes the Hamming graph, see Definition 2.3. The strategy of Alice and Bob is to agree on a graph coloring of  $\Omega_n$  before the game starts. After being given their input vector, Alice and Bob should then return the color of their vector, encoded as an  $r$ -dimensional binary vector. With this  $r$ -dimensional vector, Alice and Bob can indicate  $2^r = n$  distinct colors. Disregarding any luck in guessing, the game can always be won if and only if  $\chi(\Omega_n) \leq n$ .

The orthogonality graph gets its name from another description of the graph, that is, when the vectors have  $\{\pm 1\}$  entries. The Hamming distance between two binary vectors of  $n/2$  then corresponds to those  $\{\pm 1\}$  vectors being orthogonal to each other.

Godsil and Newman [115] prove that  $\chi(\Omega_{2^r}) = 2^r$  for  $r \in \{1, 2, 3\}$  and  $\chi(\Omega_{2^r}) > 2^r$  otherwise. This means that the game can only be won for  $r \leq 3$ . Clearly, for odd  $n$ ,  $\Omega_n$  is edgeless. We therefore restrict the analysis of  $\Omega_n$  to the case when  $n$  is a multiple of 4. When that is the case,  $\Omega_n$  consists of two isomorphic components, for vectors of even and odd Hamming weights respectively.

Next to  $\chi(\Omega_n)$ , the independence number  $\alpha(\Omega_n)$  has also been studied in multiple papers. The two graph parameters are related by  $|V| \leq \chi(G)\alpha(G)$ , for any graph  $G = (V, E)$ , see (2.73). Frankl [92] and Galliard [104] constructed a stable set of  $\Omega_n$

of size  $\underline{\alpha}(n)$  for  $n \equiv 0 \pmod{4}$ . In particular,

$$\alpha(\Omega_n) \geq \underline{\alpha}(n) := 4 \sum_{i=0}^{n/4-1} \binom{n-1}{i}.$$

On the other hand, de Klerk and Pasechnik [67] used an SDP relaxation to find  $\alpha(\Omega_{16}) = \underline{\alpha}(16) = 2306$ . It is also known that  $\alpha(\Omega_{24}) = \underline{\alpha}(24) = 178208$ , see [147]. Ihringer and Tanaka [147] proved that  $\alpha(\Omega_{2^r}) = \underline{\alpha}(2^r)$  for  $r \geq 2$ . Godsil and Newman [115] (see also [93]) conjecture that

$$\alpha(\Omega_{4m}) = \underline{\alpha}(4m) \text{ for all } m \geq 1. \quad (2.68)$$

The smallest value for which (2.68) has not been proved is  $m = 10$ . We now proceed by computing  $\vartheta_k(\Omega_n)$  when  $n$  is a multiple of four. From [237], the (unordered) eigenvalues of  $\Omega_n$  are then given by

$$\lambda_r = \frac{2^{n/2}}{(n/2)!} \prod_{i=1}^{n/2} (2i - 1 - r), \quad 1 \leq r \leq n,$$

and  $\lambda_0 = \binom{n}{n/2}$  (since  $\Omega_n$  is regular with degree  $\lambda_0$ ). The smallest eigenvalue is obtained for  $r = 2$  and thus

$$\lambda_2 = \frac{1}{1-n} \binom{n}{n/2}.$$

Since  $\Omega_n$  is isomorphic to a binary Hamming graph,  $\Omega_n$  is a symmetric graph. The bound of Theorem 2.20 thus holds with equality. The multiplicity of  $\lambda_2$  equals  $n^2 - n$  [237]. This multiplicity exceeds  $n$  since  $n$  is a multiple of 4. Then it is not hard to show (by a method comparable to the one used in the proof of Theorem 2.22) that

$$\vartheta_k(\Omega_n) = k \frac{2^n}{n}, \quad k \leq n.$$

When  $k = 1$ ,  $\vartheta_k(\Omega_n)$  coincides here with the so called ratio bound. This bound refers to (2.76) for regular graphs and was also computed for  $\Omega_n$  in [237].

Let  $S$  be a stable set of size  $\underline{\alpha}(n)$  that contains no vectors that have their Hamming weight contained in  $W := \{n/4 + 1, n/4 + 3, \dots, 3n/4 - 1\}$ . Furthermore, note that the Johnson graphs (see Definition 2.4) appear as vertex induced subgraphs of  $\Omega_n$ . Let  $w \in W$  and consider a subgraph of  $\Omega_n$  that is isomorphic to  $J(n, w, w - n/4)$ . This subgraph contains no vertices in  $S$  and thus

$$\alpha_2(\Omega_n) \geq \underline{\alpha}(n) + 4 \max_{w \in W} \{\alpha(J(n, w, w - n/4))\}.$$

We may multiply the independence number of  $J(n, w, w - n/4)$  by 4 since we can take bitwise complements and find an isomorphic stable set in the isomorphic second component of  $\Omega_n$ .

In Section 2.8.1 we prove that  $\chi_k(\Omega_{4n+2}) = 2k$ .

## 2.8 New bounds on $\chi_k(G)$

In this section we first we derive bounds on the product and sum of  $\chi_k(G)$  and  $\chi_k(\overline{G})$ . Then, we provide graphs for which the bounds are sharp. Lastly, we derive spectral lower bounds on the multichromatic number of a graph.

A famous result by Nordhaus and Gaddum [241] states that

$$\begin{aligned} n &\leq \chi(G)\chi(\overline{G}) \leq \left(\frac{n+1}{2}\right)^2, \\ 2\sqrt{n} &\leq \chi(G) + \chi(\overline{G}) \leq n+1. \end{aligned} \tag{2.69}$$

Various papers have been published on Nordhaus-Gaddum type results for other graph parameters, such as the independence and edge-independence number (see [16] for a survey). We provide Nordhaus-Gaddum type results for  $k$ -multicoloring in the following theorem.

**Theorem 2.33.** *For any graph  $G = (V, E)$ ,  $|V| = n$ , and  $k \in \mathbb{N}$ , we have*

$$\begin{aligned} k^2n &\leq \chi_k(G)\chi_k(\overline{G}) \leq k^2 \left(\frac{n+1}{2}\right)^2, \\ 2k\sqrt{n} &\leq \chi_k(G) + \chi_k(\overline{G}) \leq k(n+1). \end{aligned} \tag{2.70}$$

*Proof.* We follow the original proof as given in [241], extended to the  $k$ -multicoloring case. Let us fix some  $k \in \mathbb{N}$ , and write  $\chi = \chi_k(G)$  and  $\overline{\chi} = \chi_k(\overline{G})$ . Consider an optimal  $k$ -multicoloring of  $G$ , using  $\chi$  colors. Then for  $i = 1, 2, \dots, \chi$ , define  $n_i$  as the set of vertices that are colored with color  $i$ . We have that

$$\sum_{i=1}^{\chi} |n_i| = kn \implies \max_{i \in [\chi]} |n_i| \geq \frac{kn}{\chi}. \tag{2.71}$$

Consider a set  $n_i$  of maximum cardinality. Since the vertices in this set share a color, they form a stable set in  $G$ . Thus they form a clique in  $\overline{G}$ . Accordingly,

$$\overline{\chi} \geq k\omega(\overline{G}) \geq k \max_{i \in [\chi]} |n_i|. \tag{2.72}$$

Combining (2.71) and (2.72) proves that  $\chi\overline{\chi} \geq k^2n$ .

The lower bound on  $\chi + \overline{\chi}$  can be proven by algebraic manipulation:

$$(\chi - \overline{\chi})^2 \geq 0 \implies \chi^2 + \overline{\chi}^2 + 2\chi\overline{\chi} \geq 4\chi\overline{\chi} \implies \chi + \overline{\chi} \geq 2\sqrt{\chi\overline{\chi}} \geq 2k\sqrt{n}.$$

The two upper bounds in (2.70) follow directly from combining (2.4) and (2.69).  $\square$

The second upper bound in (2.70) can also be found in [38] (in a slightly generalized form). We now present graphs for which the bounds in Theorem 2.33 are attained. For that purpose we define the graph sum of two graphs. The graph sum of graphs  $G_1$  and  $G_2$  is the graph, denoted by  $G_1 + G_2$ , whose vertices and edges are defined as follows:

$$V(G_1 + G_2) := V(G_1) \cup V(G_2), \quad E(G_1 + G_2) := E(G_1) \cup E(G_2).$$

Nordhaus and Gaddum [241] show that the upper bounds in their theorem are attained by graph  $G = K_p + \overline{K}_{p-1}$ . Graph  $G$  has  $n = 2p - 1$  vertices. It is clear that  $\chi(G) = \chi(\overline{G}) = p = \frac{n+1}{2}$ . Thus  $G$  attains both upper bounds simultaneously. As both  $G$  and  $\overline{G}$  are weakly perfect graphs, we can apply (2.77) to find  $\chi_k(G) = \chi_k(\overline{G}) = kp = k\frac{n+1}{2}$ . This implies that graph  $G$  also attains the upper bounds in Theorem 2.33. Nordhaus and Gaddum [241] also provide an example of a graph which attains the lower bounds in their theorem. This example extends to the multichromatic variant as well. Let  $m_1 = m_2 = \dots = m_p = p$  and consider the complete multipartite graph  $G = K_{m_1, \dots, m_p}$ . Then  $\chi_k(G) = \chi_k(\overline{G}) = kp = k\sqrt{n}$ . Thus, this graph  $G$  attains the lower bounds in Theorem 2.33. In fact, for any graph  $G$  such that  $\chi(G)\chi(\overline{G}) = |V(G)|$ , we have  $k^2|V(G)| \leq \chi_k(G)\chi_k(\overline{G})$  by Theorem 2.33, and  $\chi_k(G)\chi_k(\overline{G}) \leq k^2\chi(G)\chi(\overline{G}) = k^2|V(G)|$  by (2.4). Since the upper and lower bound coincide, we have  $\chi_k(G)\chi_k(\overline{G}) = k^2|V(G)|$ . The set of vertex-transitive graphs provides a number of examples for which this bound is attained, such as the Johnson graph  $J(n, 2, 1)$  when  $n$  is even.

The chromatic number of a graph is bounded by the spectrum of matrices related to its adjacency matrix. This well-known result is given below.

**Theorem 2.34** ([142]). *If  $G$  has at least one edge, then  $\chi(G) \geq 1 - \frac{\lambda_1(A_G)}{\lambda_n(A_G)}$ .*

Since each color class has size at most  $\alpha(G)$ , we have that

$$\chi(G) \geq \frac{n}{\alpha(G)}, \quad (2.73)$$

where  $n$  is the number of vertices in  $G$ . Therefore one can use upper bounds for  $\alpha(G)$  to derive lower bounds for  $\chi(G)$ . From (2.2) it follows that  $\alpha(G \circ K_k) = \alpha(G)$ . Thus we can establish the multicoloring variant of (2.73):

$$\chi_k(G) = \chi(G \circ K_k) \geq \frac{|V(G \circ K_k)|}{\alpha(G \circ K_k)} = \frac{kn}{\alpha(G)}. \quad (2.74)$$

Note that (2.74) also follows from (2.71) (since  $\alpha(G) \geq \max |n_i| \geq kn/\chi_k(G)$ ). The bound (2.74) is also given in [47], where the authors show that the lower bound is tight for webs and antiwebs. Note that for a graph  $G$  such that  $\alpha_k(G) = k\vartheta(G)$  we have that  $\alpha_k(G) = k\alpha(G)$ , see [172, Lem. 5], and thus  $\chi_k(G) \geq \frac{k^2n}{\alpha_k(G)}$ . This inequality is satisfied by, for example, the Johnson graphs  $J(n, 2, 1)$  when  $n$  is even, and  $J(n, 3, 2)$  when  $n \equiv 1$  or  $3 \pmod 6$ , see [172, Table 1].

Let us now present known upper bounds for the independence number of a graph.

**Theorem 2.35** ([142]). *For any  $d$ -regular graph  $G$  on  $n$  vertices, we have  $\alpha(G) \leq n \frac{\lambda_n(A_G)}{\lambda_n(A_G) - d}$ .*

The result of Theorem 2.35 applies only to regular graphs with no loops. Haemers [132] generalizes the Hoffman bound as follows.

**Theorem 2.36** ([132]). *Let  $G$  have minimum vertex degree  $\delta$ . Then*

$$\alpha(G) \leq n \frac{\lambda_1(A_G)\lambda_n(A_G)}{\lambda_1(A_G)\lambda_n(A_G) - \delta^2}. \quad (2.75)$$

If  $G$  is regular, then the result of Theorem 2.36 reduces to Hoffman's bound. Another extension of the bound of Hoffman is given by Godsil and Newman [116].

**Theorem 2.37** ([116]). *Let  $G$  be a loopless graph and  $L_G$  its Laplacian matrix. Then*

$$\alpha(G) \leq n \frac{\lambda_1(L_G) - \bar{d}_G}{\lambda_1(L_G)}, \quad (2.76)$$

where  $\bar{d}_G$  denotes the average degree of the vertices of  $G$ .

Now we are ready to present our results.

**Lemma 2.38.** *Let  $G$  have minimum vertex degree  $\delta$ . Then*

$$\chi_k(G) \geq k \frac{\lambda_1(A_G)\lambda_n(A_G) - \delta^2}{\lambda_1(A_G)\lambda_n(A_G)}.$$

*Proof.* The result follows by combining (2.74) and (2.75).  $\square$

**Lemma 2.39.** *For any loopless graph  $G$ , we have*

$$\chi_k(G) \geq k \frac{\lambda_1(L_G)}{\lambda_1(L_G) - \bar{d}_G},$$

where  $\bar{d}_G$  denotes the average degree of its vertices, and  $L_G$  the Laplacian matrix of  $G$ .

*Proof.* The result follows by combining (2.74) and (2.76).  $\square$

When  $G$  is a regular graph, (2.76) is equivalent to the result of Theorem 2.35, and therefore the result of Lemma 2.39 is equivalent to:

$$\chi_k(G) \geq k \left( 1 - \frac{\lambda_1(A_G)}{\lambda_n(A_G)} \right).$$

It is not difficult to verify that complete graphs attain the bounds of Lemma 2.38 and Lemma 2.39.

We end this section by presenting bounds on the multichromatic number of Johnson graphs, see Definition 2.4. We study the simple case  $J(n, 2, 1)$ ,  $n \geq 4$ . Graph  $J(n, 2, 1)$  is sometimes referred to as the triangular graph. The graph  $J(n, 2, 1)$  is the complement graph of the Kneser graph  $K(n, 2)$ , and both are known to be strongly regular. Every vertex of  $J(n, 2, 1)$  corresponds to a set of two elements. These two elements can be thought of as two vertices of the complete graph  $K_n$ , with the vertex in  $J(n, 2, 1)$  representing the edge between these two vertices of  $K_n$ . Graph  $J(n, 2, 1)$  is thus the line graph of the complete graph  $K_n$ . For any graph  $G$ , its line graph is denoted  $L(G)$ .

**Proposition 2.40.** *For the triangular graph  $J(n, 2, 1)$ ,  $n \geq 4$ , we have*

$$k(n-1) \leq \chi_k(J(n, 2, 1)) \leq k \left( 2 \left\lfloor \frac{n-1}{2} \right\rfloor + 1 \right).$$

*Proof.* As  $J(n, 2, 1)$  is isomorphic to  $L(K_n)$ , a coloring of  $J(n, 2, 1)$  is equivalent to an edge coloring of  $K_n$ . It is not hard to see that  $\omega(L(G))$  equals the maximum degree of a vertex of  $G$ . Thus  $\omega(L(K_n)) = n - 1$ . For even  $n$ ,  $\chi(L(K_n)) = n - 1$ , see [27]. Therefore, for even  $n$  we have

$$\chi(L(K_n)) = \omega(L(K_n)) \implies \chi_k(L(K_n)) = k\chi(L(K_n)) = k(n - 1),$$

where the implication follows from (2.4). For odd  $n$ ,  $\chi(L(K_n)) = n$ , see [301]. By (2.4), the proposition follows.  $\square$

Note that in the proof of the previous proposition we could also exploit the following well-known result:  $\alpha(K(n, 2)) = n - 1$ .

### 2.8.1 Hamming graphs

In this section we present results for the (multi)chromatic number of Hamming graphs (Definition 2.3). We also provide sufficient and necessary conditions for the Hamming graph to be perfect.

In the Hamming graph  $H(n, q, F)$ , the vertex set is the set of  $n$ -tuples of letters from an alphabet of size  $q$ , and vertices  $u$  and  $v$  are adjacent if their Hamming distance satisfies  $d(u, v) \in F$ . Note that  $|V(H(n, q, F))| = q^n$ . By slight abuse of notation, we will use the terms vectors and vertices interchangeably, as they permit a one-to-one correspondence in Hamming graphs. Many authors refer to  $H(n, q, \{1\})$  as the Hamming graph. The graph  $Q_n := H(n, 2, \{1\})$  is also known as the binary Hamming graph or hypercube graph.

We first list several known results for  $H(n, q, \{1\})$ . Graph  $H(n, q, \{1\})$  equals the Cartesian product of  $n$  copies of  $K_q$ . Thus  $H(n, q, \{1\}) = \square^n K_q$ , see Definition 2.1. Furthermore, it holds  $\chi(G_1 \square G_2) = \max\{\chi(G_1), \chi(G_2)\}$ , see [267]. Therefore, the chromatic number  $\chi(H(n, q, \{1\})) = q$ . To derive the independence number of  $H(n, q, \{1\})$ , we proceed as follows. Let  $S \subset V$  be a stable set of  $H(n, q, \{1\})$ . Then  $\min_{u, v \in S, u \neq v} d(u, v) \geq 2$ . From coding theory, the Singleton bound [279] is an upper bound on the maximum number of codes of length  $n$ , using an alphabet of size  $q$ , such that a Hamming distance between any two codes is at least two. In particular, from the Singleton bound we have  $\alpha(H(n, q, \{1\})) \leq q^{n-1}$ .

To show that  $\alpha(H(n, q, \{1\})) \geq q^{n-1}$ , we construct a stable set in the Hamming graph of size  $q^{n-1}$ , by a construction employed in [279]. Consider all the vectors in  $(\mathbb{Z}/q\mathbb{Z})^n$  for which the coordinates sum to some  $x \in \mathbb{Z}/q\mathbb{Z}$ . By symmetry, there exist  $q^{n-1}$  vectors satisfying this condition. Note that any two different vectors satisfying this condition must differ in at least two positions, which implies they are not adjacent. Thus,  $\alpha(H(n, q, \{1\})) \geq q^{n-1}$  and combined with the Singleton bound, this gives  $\alpha(H(n, q, \{1\})) = q^{n-1}$ .

To the best of our knowledge, the following results are not known in the literature.

**Lemma 2.41.** *For  $k \leq q^n$ ,  $\chi_k(H(n, q, \{1\})) = kq$ .*

*Proof.* Let us denote  $H = H(n, q, \{1\})$ . Consider the vectors in  $H$  for which the first entry ranges from 0 up to and including  $q - 1$ , while the other entries equal 0. This

gives a clique of size  $q$  and since  $\omega(H) \leq \chi(H) = q$ , we have  $\omega(H) = q$ . From (2.4), it follows the result.  $\square$

The proof of Lemma 2.41 relies on the fact that  $\omega(H) = \chi(H)$ , or equivalently, that  $H(n, q, \{1\})$  is a weakly perfect graph. In general, for any weakly perfect graph  $G$

$$\omega(G) = \chi(G) \implies \chi_k(G) = k\chi(G). \quad (2.77)$$

This gives rise to the question for which values of  $q$  and  $n$  the graph  $H(n, q, \{1\})$  is perfect. The strong perfect graph theorem [57] states that a graph is perfect if and only if it does not contain  $C_{2n+1}$  or  $\overline{C}_{2n+1}$  as induced subgraphs, for all  $n > 1$ .

**Proposition 2.42.** *The Hamming graph  $H(n, q, \{1\})$  is a perfect graph if and only if  $n \leq 2$  or  $q \leq 2$ .*

*Proof.* Denote  $H(n, q) = H(n, q, \{1\})$ . Graph  $H(1, q)$  is  $K_q$ , which is clearly a perfect graph. Graph  $H(2, q)$  is a lattice graph, or Rook's graph, which is also a perfect graph. Graph  $H(n, 1)$  is a single vertex and thus also perfect. Lastly, graph  $H(n, 2)$  is bipartite and thus perfect. For  $q \geq 3$ , the following vectors from  $H(3, q)$  form  $C_7$ :

$$\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}, \begin{bmatrix} 2 \\ 1 \\ 1 \end{bmatrix}, \begin{bmatrix} 2 \\ 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}. \quad (2.78)$$

Then by the strong perfect graph theorem,  $H(3, q)$  is not perfect. An odd cycle in  $H(n, q)$  for general  $n, q \geq 3$ , is obtained by simply adjoining zeros to the vectors in (2.78) such that they become  $n$ -dimensional.  $\square$

As  $H(n, q, \{f\})$  is edgeless for  $f > n$ , we consider the extremal case  $H(n, q, \{n\})$  for  $n > 1$ . Note that  $H(1, q, \{1\}) = K_q$ . Graph  $H(n, q, \{n\})$  can be described by use of the tensor product of graphs (see Definition 2.1). In particular, we have that  $H(n, q, \{n\}) = \otimes^n K_q$ .

Since all the edges of  $G_1 \otimes G_2$  also appear in  $G_1 \circ G_2$ , it follows that  $\chi(G_1 \otimes G_2) \leq \chi(G_1 \circ G_2)$ . Moreover, by [137], we have

$$\chi(G_1 \otimes G_2) \leq \min\{\chi(G_1), \chi(G_2)\}. \quad (2.79)$$

*Hedetniemi's conjecture* [137] states that (2.79) holds with equality. The conjecture was recently disproved by Shitov [275]. Inequality (2.79) implies that  $\chi(\otimes^n K_q) \leq q$ .

The vectors  $i \cdot \mathbf{1}$  for  $0 \leq i \leq q - 1$  form a clique of size  $q$  in graph  $H(n, q, \{n\})$ . Thus  $q \leq \omega(H(n, q, \{n\}))$ . Now, from this inequality and  $\chi(\otimes^n K_q) \leq q$  it follows that  $\chi(H(n, q, \{n\})) = q$ . Using (2.4) or (2.77), we find  $\chi_k(H(n, q, \{n\})) = kq$ . The coloring of these tensor products of graphs has been previously considered by Greenwell and Lovász [122], where they also proved this result.

Let us now define  $f^+ := \{i \in \mathbb{N} : f \leq i\}$ . The Hamming graph  $H(n, q, f^+)$  has been studied by El Rouayheb et al. [82] among others. They show that, under some condition on the parameters  $n, q$  and  $f$ ,  $\chi(H(n, q, f^+)) = q^{n-f+1}$ . We extend this result to multicoloring in the following proposition.

**Proposition 2.43.** *For  $q \geq n - f + 2$  and  $1 \leq f \leq n$ , we have that  $\chi_k(H(n, q, f^+)) = kq^{n-f+1}$ .*

*Proof.* For parameters  $n$ ,  $q$  and  $f$  satisfying the conditions of the proposition, it is known (cf. [94]) that  $\alpha(H(n, q, f^+)) = q^{f-1}$ . By (2.4) and (2.74), the proposition follows.  $\square$

Binary Hamming graphs  $H(n, 2, \{f\})$ ,  $f \leq n$  form another interesting case. Recall that the Hamming weight of a vector is its Hamming distance to the zero vector, and that the Hamming graphs are vertex-transitive.

**Theorem 2.44.** *For all  $n \in \mathbb{N}$ ,  $f$  odd and  $f \leq n$ ,  $\chi_k(H(n, 2, \{f\})) = 2k$ .*

*Proof.* Let  $n, f \in \mathbb{N}$ ,  $f$  odd and  $f \leq n$ . Consider the zero vector in  $H(n, 2, \{f\})$ . Note that every vector adjacent (orthogonal) to the zero vector has an odd Hamming weight. By vertex transitivity, all vectors of even Hamming weight only have vectors of odd Hamming weight as neighbors. Similarly, vectors of odd Hamming weight only have vectors of even Hamming weight as neighbors. Graph  $H(n, 2, \{f\})$  is thus bipartite, which, combined with (2.4), proves the theorem.  $\square$

We can use Theorem 2.44 to compute the multichromatic number of certain orthogonality graphs, defined in Section 2.7.

**Corollary 2.45.** *Let  $\Omega_{4n+2}$  ( $n \in \mathbb{N}$ ) be the orthogonality graph. Then,*

$$\chi_k(\Omega_{4n+2}) = 2k.$$

*Proof.* Graph  $\Omega_{4n+2}$  is isomorphic to  $H(4n+2, 2, \{2n+1\})$ . This corollary is thus a special case of Theorem 2.44.  $\square$

## 2.9 Conclusions

In this chapter, we study the generalized  $\vartheta$ -number for highly symmetric graphs and beyond. The parameter  $\vartheta_k(G)$  generalizes the concept of the famous  $\vartheta$ -number that was introduced by Lovász [204]. Since  $\vartheta_k(G)$  is sandwiched between  $\alpha_k(G)$  and  $\chi_k(\overline{G})$ , it serves as a bound for both graph parameters.

Several results in this chapter are not restricted to highly symmetric graphs. In particular, the results in Sections 2.2 to 2.4. In Section 2.2 we present in an elegant way a known result that  $\vartheta_k(G)$  is a lower bound on  $\chi_k(\overline{G})$ . Another lower bound on  $\chi_k(\overline{G})$  is  $k\vartheta(G)$ , see (2.8). The inequality (2.8) is rather counter-intuitive since it is more difficult to compute  $\vartheta_k(G)$  than  $\vartheta(G)$ , while  $k\vartheta(G)$  provides a better bound on the  $k$ th chromatic number. However, the generalized  $\vartheta$ -number can also be used to compute lower bounds for the (classical) chromatic number of a graph, see Section 2.5.2.

In Section 2.3 we show that the sequence  $(\vartheta_k(G))_k$  is increasing and bounded above by the order of  $G$  (Proposition 2.8 and Theorem 2.10), and that the increments of the sequence can be arbitrarily small (Theorem 2.11). Section 2.4 provides bounds for  $\vartheta_k(G)$  where  $G$  is the strong graph product of two graphs (Theorem 2.13) and the disjunction product of two graphs (Theorem 2.14).

Sections 2.5 to 2.7 consider highly symmetric graphs. We derive closed form expressions for the generalized  $\vartheta$ -number on cycles (Theorem 2.17), Johnson (Theorem 2.22), Kneser (Corollary 2.24), and strongly regular graphs (Theorem 2.29), among other results. It is known that  $\vartheta(K_k \square G)$  and  $\vartheta_k(G)$  provide upper bounds on  $\alpha_k(G)$ . However, it is more computationally demanding to compute  $\vartheta(K_k \square G)$  than  $\vartheta_k(G)$ . We show that for graphs that are both edge-transitive and vertex-transitive it suffices to solve  $\vartheta_k(G)$ , see Theorem 2.28. However, the gap between  $\vartheta_k(G)$  and  $\vartheta(K_k \square G)$  can be arbitrarily large (Proposition 2.27). We also prove that  $\vartheta'(K_k \square G)$  equals  $\vartheta'_k(G)$  for any (non-trivial) strongly regular graph  $G$  and  $k < n(r+1)/(r+n-d)$ , see Theorem 2.32. Section 2.7 presents results for  $\vartheta_k(G)$  and  $\chi_k(G)$  on orthogonality graphs.

Bounds on the  $k$ th chromatic number of various graphs are given in Section 2.8. In particular, bounds on the product and sum of  $\chi_k(G)$  and  $\chi_k(\overline{G})$  are presented in Theorem 2.33. Lemma 2.41, Proposition 2.43, and Theorem 2.44 provide the multichromatic number for several Hamming graphs, while Proposition 2.40 provides bounds for the multichromatic number on triangular graphs.

We list two open problems. The first one is to prove Conjecture 2.25 for any graph. Recall that we prove Conjecture 2.25 only for symmetric graphs, see Theorem 2.28. It would be interesting to prove the conjecture by Godsil, Newman and Frankl (2.68) for the first open case  $\Omega_{40}$ . The second open problem is to generalize the well-known inequality  $\vartheta(G)\vartheta(\overline{G}) \geq |V|$ , see [204], for  $\vartheta_k(G)$ ,  $k \geq 2$ .

### 3 Cuts and semidefinite liftings for the complex cut polytope

The max-cut problem on a graph is to find a partition of its vertices in two disjoint subsets, that maximizes the number of edges that cross the partition (see also Section 1.3.2). The max-cut problem finds applications in VLSI design and physics [26], data science [75], and is NP-hard. The convex hull of the rank 1 matrices representing all partitions is known as the cut polytope. This polytope admits an exponential number (in  $n$ ) of extreme points, and it cannot be efficiently described, in contrast to its positive semidefinite (PSD) approximation, the elliptope [186].

We consider here complex generalizations of the cut polytope and elliptope, namely the complex cut polytope, denoted  $\text{CUT}_m^n$ , and the complex elliptope, denoted  $\mathcal{E}_m^n$ . For fixed integers  $m$  and  $n$ ,  $\text{CUT}_m^n$  is defined as the convex hull of Hermitian rank 1 matrices  $xx^H$ , where the elements of the vectors  $x \in \mathbb{C}^n$  are  $m$ th unit roots. For  $m = 2$ ,  $\text{CUT}_m^n$  corresponds to the cut polytope. The set  $\text{CUT}_m^n$  finds applications in the multiple-input multiple-output detection problem (MIMO) [151, 208, 226, 317], angular synchronization [24], phase retrieval [303], radar signal processing [209, 287], and for  $m = 3$ , it can be used to model the max-3-cut problem [118]. For finite  $m \geq 3$ , algorithms for optimization over  $\text{CUT}_m^n$  are proposed in [207, 209], and approximation ratios are studied in [286, 316].

In this chapter, we derive novel cuts in the complex plane that separate  $\mathcal{E}_m^n$  from  $\text{CUT}_m^n$ . In particular, we derive all facets of  $\text{CUT}_3^3$  to obtain an exact description. We define a function  $\text{str}$ , that provides the approximation ratio of maximization over  $\mathcal{E}_m^n$  and maximization over  $\text{CUT}_m^n$ , for given problem instances. This function is used for numerically evaluating the effect of adding valid cutting planes to  $\mathcal{E}_m^n$ . We prove that the cuts introduced here are invariant under rotations and taking the complex conjugate. We also investigate the effect of adding cuts to  $\mathcal{E}_m^n$  for various optimization problems.

Optimization over  $\mathcal{E}_m^n$  can be done in polynomial time (for fixed precision), by solving a complex semidefinite program (CSDP). CSDPs have recently received much attention in the literature [154, 213, 304, 305, 306, 319]. CSDPs with matrix variables of order  $n$  are solved by SDP solvers as real SDPs with matrix variables of order  $2n$ . In [76, Corollary 2.5.2] and [306], conditions are provided under which this doubling of the size can be avoided. In this chapter, we extend these conditions. Specifically, we show that CSDPs can be reformulated as real SDPs of same size, when the objective function contains only real coefficients. In particular, we show that the facet defining

inequalities of  $\mathcal{E}_m^n$  can be equivalently reformulated to real facet defining inequalities.

The set  $\text{CUT}_\infty^n$  is studied in [150]. The first semidefinite lifting of  $\text{CUT}_\infty^n$ , denoted  $\mathcal{E}_\infty^n$ , is also known as the set of correlation matrices [126, 194]. Here, we extend the results of [150]. In particular, we consider second semidefinite Lasserre-type liftings of  $\text{CUT}_\infty^n$ . Such liftings are defined in terms of moment matrices, and we study second liftings with smaller moment matrices than those proposed in the literature [150, 154]. Despite this decrease in size, we show that here considered liftings are equivalent to those proposed in the literature. Moreover, for  $n = 4$  (the smallest  $n$  for which  $\text{CUT}_\infty^n \subsetneq \mathcal{E}^n$ ), we prove that the second semidefinite lifting of  $\text{CUT}_\infty^n$  excludes all rank 2 extreme points present in  $\mathcal{E}_\infty^4$ , and that matrices in this set satisfy a certain valid cut for  $\text{CUT}_\infty^4$ .

We also show, via a constructive proof, that  $\mathcal{E}_m^n$  contains rank 2 extreme points for all integer  $n, m \geq 3$ . This shows the strict inclusion of  $\text{CUT}_m^n$  in  $\mathcal{E}_m^n$  for these values of  $n$  and  $m$ . For  $n = 3$ , we provide necessary and sufficient conditions for matrices to be rank 2 extreme points of  $\mathcal{E}_m^3$ .

This chapter is organized as follows. Section 3.1 provides preliminaries. Section 3.2, introduces a framework for finding valid inequalities for  $\text{CUT}_m^n$  and provide some valid cuts. In Section 3.3, we provide an exact description of  $\text{CUT}_3^3$ , and use the derived facets of  $\text{CUT}_3^3$  to strengthen  $\mathcal{E}_3^n$  for general  $n \geq 3$ . In Section 3.4 we provide an efficient reformulation of CSDPs whose convex feasible sets are closed under complex conjugation. In Section 3.5 we investigate the sets  $\text{CUT}_\infty^n$  and second semidefinite liftings of  $\text{CUT}_\infty^n$ . In Section 3.6, we study rank 2 extreme points of  $\mathcal{E}_m^n$  for integer  $m > 2$ . In Section 3.7, we numerically investigate the effect of adding cuts to  $\mathcal{E}_m^n$  for various optimization problems from the literature. Lastly, in Section 3.8 we draw conclusions, and propose future research directions.

## 3.1 Preliminaries

We restate the definitions from Section 1.3.2.1. We define, for fixed integer  $m \geq 2$ , the set

$$\mathcal{U}_m := \left\{ e^{\theta \mathbf{i}} : \theta = \frac{2\pi k}{m}, k \in [m] \right\} \subseteq \mathbb{C}, \quad (3.1)$$

where  $\mathbf{i} := \sqrt{-1}$  denotes the imaginary unit. Note that  $\mathcal{U}_m$  is the set of the complex  $m$ th roots of unity. We define  $\mathcal{U}_m^n$  as the set containing  $m^n$  vectors of length  $n$ , in which each entry is restricted to be an element of  $\mathcal{U}_m$ .

In this chapter, we consider a generalization of the well-known cut polytope [74], which we refer to as the *complex cut polytope*. For integers  $n, m \geq 2$ , the complex cut polytope is defined as

$$\text{CUT}_m^n := \text{Conv} \{ xx^{\text{H}} : x \in \mathcal{U}_m^n \}, \quad (3.2)$$

where  $(\cdot)^{\text{H}}$  is the Hermitian transpose, also known as conjugate transpose. As  $\mathcal{U}_2 = \{\pm 1\}$ , the set  $\text{CUT}_2^n$  coincides with the well-known cut polytope, which is a feasible

set for the max-cut problem [117, 186]. Optimization problems over  $\text{CUT}_m^n$ ,  $m \geq 2$ , are NP-hard, as they include the NP-hard max-cut problem.

A matrix  $X \in \mathbb{C}^{n \times n}$  is said to be Hermitian if  $X^H = X$ . Let  $\mathcal{H}^n \subseteq \mathbb{C}^{n \times n}$  be the set of  $n \times n$  Hermitian matrices. For  $X \in \mathcal{H}^n$ , if  $v^H X v \geq 0$  for all  $v \in \mathbb{C}^n$ , we say that  $X$  is (Hermitian) PSD. Let  $\mathcal{H}_+^n$  be the set of  $n \times n$  Hermitian PSD matrices. We define *the complex elliptope* as follows:

$$\mathcal{E}_m^n := \{X \in \mathcal{H}_+^n : \text{diag}(X) = \mathbf{1}, X_{ij} \in \text{Conv}(\mathcal{U}_m) \forall i, j \in [n]\}. \quad (3.3)$$

Note that for  $m = 2$ , the constraints  $X_{ij} \in \text{Conv}(\mathcal{U}_2) = [-1, 1]$  are redundant, as they are implied by  $X \succeq 0$  and  $\text{diag}(X) = \mathbf{1}$ . Thus, the complex elliptope  $\mathcal{E}_2^n$  corresponds to the elliptope that is defined by Laurent and Poljak [186]. For  $m = 3$ , the complex elliptope  $\mathcal{E}_m^n$  corresponds to the feasible set of the CSDP relaxation for the max-3-cut problem by Goemans and Williamson [118]. It is clear that  $\text{CUT}_m^n \subseteq \mathcal{E}_m^n$ .

Here, we derive strong approximations of  $\text{CUT}_m^n$  by using CSDP. Besides considering second semidefinite liftings, we also derive cuts in the complex plane that separate  $\mathcal{E}_m^n$  from  $\text{CUT}_m^n$ . Cuts in the complex plane have recently been studied by Jarre et al. [150], for the set  $\text{CUT}_\infty^n$ , defined as

$$\text{CUT}_\infty^n := \text{Conv} \{xx^H : x \in \mathbb{C}^n, |x_i| = 1 \forall i \in [n]\}, \quad (3.4)$$

where  $|\cdot|$  denotes the absolute value of a complex number. That is, for  $z \in \mathbb{C}$ ,  $|z| := \sqrt{z\bar{z}}$ , where  $\bar{z}$  is the complex conjugate of  $z$ . We also define  $\mathcal{U}_\infty := \{\exp(\theta \mathbf{i}) : \theta \in \mathbb{R}\}$ , for  $\exp(\cdot)$  the exponential function, as a natural extension of (3.1), and the complex elliptope

$$\mathcal{E}_\infty^n := \{X \in \mathcal{H}_+^n : \text{diag}(X) = \mathbf{1}_n\}.$$

Note that for  $X \in \mathcal{E}_\infty^n$ , we have  $X_{ij} \in \text{Conv}(\mathcal{U}_\infty) = \{x \in \mathbb{C} : |x| \leq 1\}$ . The complex elliptope  $\mathcal{E}_\infty^n$  can be considered as the first semidefinite lifting of  $\text{CUT}_\infty^n$ . Additionally, one can define a second semidefinite lifting of  $\text{CUT}_\infty^n$ , following [175], and also proposed by Jarre et al. [150] for  $n = 4$ .

For any  $z \in \mathbb{C}$ ,  $\text{Re}(z) \in \mathbb{R}$  and  $\text{Im}(z) \in \mathbb{R}$  denote the real and imaginary part of  $z$ , respectively. Additionally, by slight abuse of notation, for sets  $U \subseteq \mathbb{C}^n$ , we define  $\text{Re}(U) := \{\text{Re}(u) : u \in U\} \subseteq \mathbb{R}^n$ . The boundary of a set  $U$  is denoted by  $\partial U$ .

**Remark 3.1.** The set  $\text{CUT}_m^n$  can be equivalently stated as

$$\text{CUT}_m^n = \text{Conv} \left\{ zz^H : z = \begin{bmatrix} 1 \\ x \end{bmatrix}, x \in \mathcal{U}_m^{n-1} \right\}. \quad (3.5)$$

Indeed, we may write any  $x \in \mathcal{U}_m^n$  as  $x = [x_1 \ y^\top]^\top \in \mathcal{U}_m^n$  for some  $x \in \mathcal{U}_m$  and  $y \in \mathcal{U}_m^{n-1}$ . Define  $z := \bar{x}_1 x = [1 \ \bar{x}_1 y^\top]^\top$  and note that  $z \in \mathcal{U}_m^n$ , and  $zz^H = (\bar{x}_1 x)(\bar{x}_1 x)^H = |x_1|^2 xx^H = xx^H$ . The set  $\text{CUT}_\infty^n$  can also be rewritten in a manner similar to (3.5).  $\triangle$

### 3.1.1 Basic CSDP relaxations

In this section, we present the basic semidefinite program whose feasible set is the complex elliptope  $\mathcal{E}_m^n$  for integer  $m \geq 2$ , see (3.3). The basic SDP relaxation for  $m = 2$  was introduced by Goemans and Williamson [117], for  $m = 3$  by Goemans and Williamson [118], and for general  $m \geq 3$  by Lu et al. [207]. In the sections that follow, we will derive cuts that strengthen the basic SDP. Let  $n, m \geq 2$  and  $C \in \mathcal{H}^n$ . From the definitions of  $\text{CUT}_m^n$  and  $\mathcal{E}_m^n$ , we have that  $\max_{x \in \mathcal{U}_m^n} x^H C x = \max_{X \in \text{CUT}_m^n} \langle C, X \rangle$ , and that this value upper bounded by

$$\max_{X \in \mathcal{E}_m^n} \langle C, X \rangle. \quad (\text{CSDP-P})$$

The program CSDP-P (for complex semidefinite program in primal form) can be solved in polynomial time up to desired accuracy by the interior-point method. Note that CSDP-P is strictly feasible since  $\mathcal{E}_m^n$  contains positive definite matrices, e.g., the identity.

For  $X \in \mathcal{E}_m^n$ , we require that  $X_{ij} \in \text{Conv}(\mathcal{U}_m)$ , see (3.1). One way to enforce this is to set

$$X_{ij} = \sum_{k=1}^m \lambda_k e^{\theta_k \mathbf{i}}, \quad \text{with} \quad \sum_{k=1}^m \lambda_k = 1, \quad \lambda \geq 0, \quad \lambda \in \mathbb{R}^m \quad \text{and} \quad \theta_k = \frac{2\pi k}{m},$$

i.e.,  $\exp(\theta_k \mathbf{i}) \in \mathcal{U}_m$ . Alternatively,  $X_{ij}$  can be restricted to lie in the intersection of certain half-spaces. This perspective follows from the well-known fact that  $\mathbb{C}$  is isomorphic to  $\mathbb{R}^2$  via the bijective mapping

$$g : \mathbb{R}^2 \rightarrow \mathbb{C}, \quad g(a) = a_1 + a_2 \mathbf{i},$$

and that, for  $a, b \in \mathbb{R}^2$ ,  $a^\top b = \text{Re}(\overline{g(a)}g(b))$ . Now, it is easy to see that the set  $\text{Conv}(\mathcal{U}_m)$  is given by an  $m$ -sided regular convex polygon in  $\mathbb{C}$ . For the edge connecting  $\exp(\theta_k \mathbf{i})$  and  $\exp(\theta_{k-1} \mathbf{i})$ , its normal vector (complex number) is given by  $\nu_k := \exp[(\theta_k + \theta_{k-1})\mathbf{i}/2] = \exp[(2k-1)\pi\mathbf{i}/m]$  for  $k \in [m]$ . Thus,

$$X_{ij} \in \text{Conv}(\mathcal{U}_m) \iff \text{Re}(\bar{\nu}_k X_{ij}) \leq \cos\left(\frac{\pi}{m}\right) \quad \forall k \in [m], \quad (3.6)$$

see also [207].

**Remark 3.2.** The set  $\text{Conv}(\mathcal{U}_m)$  is the regular  $m$ -sided polygon in  $\mathbb{C}$ , and as (3.6) shows, this set can be described by its  $m$  facets. A more efficient encoding can be derived by viewing  $\text{Conv}(\mathcal{U}_m)$  as the projection of a higher dimensional polytope with possibly fewer facets. The earliest result in this direction is due to Ben-Tal and Nemirovski [28]. They showed that, for  $m$  a power of two,  $\text{Conv}(\mathcal{U}_m)$  is equivalent to the projection of a polytope with  $\mathcal{O}(\log m)$  number of facets. Later results extended this to general  $m$  [89, Thm. 2], and projections of higher dimensional spectrahedra [86, 87]. Since  $m$  is small in our numerical experiments, we do not use such liftings of the regular polygon in our CSDP relaxations.  $\triangle$

To state (3.6) in terms of matrix inner products, we define, for  $k \in [m]$ ,  $i, j \in [n]$ ,  $i < j$  the Hermitian matrices

$$W_{ij}^k := \frac{1}{2} (\nu_k E_{ij} + \bar{\nu}_k E_{ji}).$$

Here,  $E_{ij}$  is the matrix that is zero everywhere, except for entry  $(i, j)$ , which has value 1. Matrix  $E_{ji}$  is defined similarly. It follows that  $\operatorname{Re}(\bar{\nu}_k X_{ij}) = \langle W_{ij}^k, X \rangle$ . Now, from SDP duality theory, it follows that the corresponding dual problem of CSDP-P is given by

$$\begin{aligned} \min \mathbf{1}^\top \mu + \cos\left(\frac{\pi}{m}\right) \sum_{ij \in [n]^2, i < j, k \in [m]} \omega_{ij}^k \\ \text{s.t. } \operatorname{Diag}(\mu) + \sum_{ij \in [n]^2, i < j, k \in [m]} \omega_{ij}^k W_{ij}^k - C \succeq 0 \quad (\text{CSDP-D}) \\ \mu \in \mathbb{R}^n, \omega_{ij} = (\omega_{ij}^1, \dots, \omega_{ij}^m)^\top \in \mathbb{R}_+^m, \quad \forall i, j \in [n], i < j. \end{aligned}$$

One can strengthen CSDP-P and CSDP-D ( $D$  for dual) via the moment and sum of squares hierarchies by Lasserre [175]. We consider this in more detail in Section 3.5, where we consider a second semidefinite lifting of  $\text{CUT}_m^4$ . In Section 3.2 we strengthen CSDP-P by adding valid cuts to  $\mathcal{E}_m^n$ , which can be considered as the first semidefinite lifting of  $\text{CUT}_m^n$ .

## 3.2 Framework for finding valid inequalities for $\text{CUT}_m^n$

In this section we introduce a general framework to derive valid inequalities for  $\text{CUT}_m^n$ , see (3.2). These inequalities can be then used to strengthen CSDP-P. We show that these inequalities can be classified in equivalence classes, which we derive from the group structure of  $\text{CUT}_m^n$ .

Similar to the classical result by Rockafellar [265, Thm. 18.8], stating that any real closed convex set is the intersection of all its half-spaces containing it, the set  $\text{CUT}_m^n$  has an equivalent description as follows.

**Proposition 3.3.**

$$\text{CUT}_m^n = \left\{ X \in \mathcal{H}^n : \langle Q, X \rangle \leq \max_{x \in \mathcal{U}_m^n} x^\text{H} Q x, \forall Q \in \mathcal{H}^n \right\}. \quad (3.7)$$

The proof of Proposition 3.3 is similar to the proof of Rockafellar [265, Thm. 18.8] and therefore omitted. Observe that, since  $Q$  is Hermitian, the values  $\langle Q, X \rangle$  and  $\max_{x \in \mathcal{U}_m^n} x^\text{H} Q x$  are real. Therefore, the inequalities in (3.7) are well-defined.

Let us exploit the formulation of  $\text{CUT}_m^n$  given by (3.7) for deriving cuts that can be added to CSDP-P in order to improve that relaxation. We define the function  $\text{str} : \mathcal{H}^n \setminus \{\mathbf{0}\} \times \mathbb{N} \rightarrow [1, \infty)$  ( $\text{str}$  for strength), as follows:

$$\text{str}(Q, m) := \frac{\max_{X \in \mathcal{E}_m^n} \langle Q, X \rangle}{\max_{X \in \text{CUT}_m^n} \langle Q, X \rangle}. \quad (3.8)$$

We assume w.l.o.g. that both the numerator and denominator in fraction (3.8) are strictly positive. This is the case for all here considered matrices  $Q$ . Clearly, for a matrix  $Q$  for which  $\max_{X \in \text{CUT}_m^n} \langle Q, X \rangle \leq 0$ , one can appropriately adjust its diagonal.

Observe that  $\text{str}$  returns the approximation ratio of maximization over  $\mathcal{E}_m^n$  and maximization over  $\text{CUT}_m^n$ , for a specific problem instance given by  $Q$  (see also [186, Section 4]). Since  $\max_{X \in \mathcal{E}_m^n} \langle Q, X \rangle$  is an upper bound on  $\max_{X \in \text{CUT}_m^n} \langle Q, X \rangle$  (both these values are assumed to be strictly positive), we have that  $\text{str}(Q, m) \geq 1$ . To improve the quality of this upper bound, one can find valid inequalities for  $\text{CUT}_m^n$ , that are violated by  $\arg \max_{X \in \mathcal{E}_m^n} \langle Q, X \rangle$ . Thus, if  $\text{str}(Q, m) > 1$ , then by adding the cut

$$\langle Q, X \rangle \leq \max_{X \in \text{CUT}_m^n} \langle Q, X \rangle, \quad (3.9)$$

to CSDP-P one may strengthen that relaxation. Note that it is, in general, NP-hard to compute  $\text{str}(Q, m)$ . However, for some  $Q$  we can find optimal solutions of both maximization problems in (3.8) analytically, and thus evaluate  $\text{str}(Q, m)$ , see Section 3.2.3.

**Remark 3.4.** For any  $c \in \mathbb{R}_+^n$ ,  $1 \leq \text{str}(Q + \text{Diag}(c), m) \leq \text{str}(Q, m)$ . In order to fairly compare cuts, we consider matrices  $Q$  that satisfy  $\langle Q, \mathbf{I} \rangle = 0$ .  $\triangle$

### 3.2.1 Symmetries of $\text{CUT}_m^n$

In this section, we formally specify symmetries of the set  $\text{CUT}_m^n$ , for finite  $m$ , that follow from the underlying group structure of  $\mathcal{U}_m$ . These symmetries will be exploited later in Sections 3.2.2 and 3.3.

As starting point, consider the set  $\mathcal{U}_m$  defined in (3.1). With the usual multiplication of complex numbers,  $\mathcal{U}_m$  forms a cyclic group of order  $m$  with identity element 1. The set  $\mathcal{U}_m^n$  with the Hadamard product forms a finite abelian group, with identity element  $\mathbf{1}_n$ . Indeed, if  $x \in \mathcal{U}_m^n$ , its inverse element is given by  $\bar{x}$ , since  $\bar{x} \odot x = \mathbf{1}_n$ .

We define, for  $\alpha \in \mathcal{U}_m^n$ , the linear group action of  $f_\alpha : \mathcal{H}^n \rightarrow \mathcal{H}^n$  as  $f_\alpha(Z) = (\alpha \alpha^H) \odot Z = \text{Diag}(\alpha) Z \text{Diag}(\bar{\alpha})$ . To see that  $f_\alpha$  defines a group action, let  $Z \in \mathcal{H}^n$  and note that  $f_{\mathbf{1}_n}(Z) = \mathbf{J}_n \odot Z = Z$ , and

$$f_\alpha(f_\beta(Z)) = (\alpha \alpha^H) \odot (\beta \beta^H) \odot Z = (\alpha \odot \beta) (\alpha \odot \beta)^H \odot Z = f_{\alpha \odot \beta}(Z). \quad (3.10)$$

Since  $f_\alpha$  defines a group action,  $f_\alpha$  is also invertible, with inverse  $f_\alpha^{-1} = f_{\bar{\alpha}}$ , which follows from (3.10) by taking  $\beta = \bar{\alpha}$ . That is,  $f_\alpha(f_{\bar{\alpha}}(Z)) = f_{\alpha \odot \bar{\alpha}}(Z) = f_{\mathbf{1}_n}(Z) = Z$ .

Note that for  $Z, Z' \in \mathcal{H}^n$ , we have

$$\langle f_\alpha(Z), f_\alpha(Z') \rangle = \langle Z, Z' \rangle. \quad (3.11)$$

Informally, function  $f_\alpha$  can be considered as acting on  $\mathcal{H}^n$  by applying some rotation to the elements of its input matrix. The number of such rotations is equal to the number of different functions  $f_\alpha$ . This number is given by  $m^{n-1}$ , and not by  $|\mathcal{U}_m^n| = m^n$ , because  $f_\alpha = f_{x\alpha}$  for any  $x \in \mathcal{U}_m$  and  $\alpha \in \mathcal{U}_m^n$ . Therefore, in the context of  $f_\alpha$ , one element of  $\alpha$  can be assumed fixed, say  $\alpha_1 = 1$ .

The sets  $\text{CUT}_m^n$  and  $\mathcal{E}_m^n$  are closed under the group action  $f_\alpha$ . To see this for  $\text{CUT}_m^n$ , note that  $f_\alpha(xx^H) \in \text{CUT}_m^n$  for  $x \in \mathcal{U}_m^n$ . Therefore, the action  $f_\alpha$  on a convex combination of rank one matrices in  $\text{CUT}_m^n$  returns a convex combination of (possibly different) rank one matrices in  $\text{CUT}_m^n$ . To see that  $\mathcal{E}_m^n$  is also closed under  $f_\alpha$ , note that the Hadamard product of PSD matrices is again PSD, due to the well-known Schur product theorem. Moreover, the elements of  $f_\alpha(X)$ ,  $X \in \mathcal{E}_m^n$ , are contained in  $\text{Conv}(\mathcal{U}_m)$ , since  $\text{Conv}(\mathcal{U}_m)$  itself is also closed under the Hadamard product. Specifically,  $(f_\alpha(X))_{ij} = X_{ij}\alpha_i\bar{\alpha}_j \in \text{Conv}(\mathcal{U}_m)$ .

The sets  $\text{CUT}_m^n$  admit two additional symmetries. These symmetries are induced permutation of rows and columns, and by complex conjugation. For the permutation symmetry, we define, for any  $X \in \mathcal{H}^n$  and permutation  $\sigma : [n] \rightarrow [n]$ , the matrix

$$X_\sigma = (X_{\sigma(i),\sigma(j)})_{i,j \in [n]}, \text{ for a permutation } \sigma : [n] \rightarrow [n], \text{ and } X \in \mathcal{H}^n. \quad (3.12)$$

Note that  $X_\sigma$  is the matrix obtained after permuting the rows and columns of  $X$  according to  $\sigma$ . We have that  $X \in \text{CUT}_m^n$  if and only if  $X_\sigma \in \text{CUT}_m^n$  for any permutation  $\sigma$ .

For complex conjugation, we have that  $X \in \text{CUT}_m^n$  if and only if  $\bar{X} \in \text{CUT}_m^n$ . More generally,  $X \rightarrow \bar{X}$  defines a reflection symmetry of  $\text{CUT}_m^n$ . Complex conjugation also defines one of the reflection symmetries of  $\mathcal{U}_m$ . In Appendix A.5 on Page 200, we show that all reflection symmetries  $\sigma$  of  $\mathcal{U}_m$  can be extended to  $\text{CUT}_m^n$ , but that  $\sigma(X) = \bar{X}$  for all reflections  $\sigma$  and  $X \in \text{CUT}_m^n$ . Stated differently, any reflection symmetry of  $\mathcal{U}_m$ , reduces to complex conjugation when extended to  $\text{CUT}_m^n$ .

**Remark 3.5.** We have not explicitly covered the symmetries of  $\text{CUT}_\infty^n$ , see (3.4). However, the group structures of  $\mathcal{U}_\infty$  and  $\mathcal{U}_\infty^n$  are similar to those of  $\mathcal{U}_m$  and  $\mathcal{U}_m^n$  for  $m$  finite, and therefore do not warrant special consideration. In particular,  $\mathcal{U}_\infty^n$  with the Hadamard product is abelian like  $\mathcal{U}_m^n$ , although its order is infinite, in contrast to  $\mathcal{U}_m^n$ . Also the group action  $f_\alpha$  for  $\alpha \in \mathcal{U}_\infty^n$  is defined similarly as  $f_\alpha$  for  $\alpha \in \mathcal{U}_m^n$ , and there is an infinite number of such group actions.

In addition,  $\text{CUT}_\infty^n$  is also closed under complex conjugation and permutation of rows and columns.  $\triangle$

### 3.2.2 Classes of valid inequalities

We show here that the strength of a valid inequality, generated by  $Q \in \mathcal{H}_n$ , is invariant under rotation of elements in  $Q$ , i.e.,  $f_\alpha(Q)$ , and taking the complex conjugate of  $Q$ . Thus, each  $Q$  in (3.9) induces a class of valid inequalities.

Consider, for  $\text{CUT}_2^n$ , the triangle inequalities [180], given by

$$c_1X_{ij} + c_2X_{ik} + c_3X_{jk} \geq -1, \quad c \in \{\pm 1\}^3, \quad c_1c_2c_3 = 1. \quad (3.13)$$

There are four ways to choose the vector  $c$ , and we say that triangle inequalities induced by different  $c$  are equivalent under rotation of coefficients (*ROC equivalent*). We generalize the notion of ROC equivalence to  $\text{CUT}_m^n$ ,  $m \geq 2$ , see also [150], by using the symmetries of  $\text{CUT}_m^n$ , as outlined in Section 3.2.1.

**Lemma 3.6.** *Let  $m, n \geq 2$  be integer numbers,  $Q \in \mathcal{H}^n$ , and  $\alpha \in \mathcal{U}_m^n$ . Then*

$$\text{str}(Q, m) = \text{str}((\alpha\alpha^H) \odot Q, m),$$

see (3.8). We say that the cuts induced by  $Q$  and  $(\alpha\alpha^H) \odot Q$  are ROC equivalent.

*Proof.* It follows from (3.11) that

$$\max_{X \in \text{CUT}_m^n} \langle Q, X \rangle = \max_{X \in \text{CUT}_m^n} \langle f_\alpha(Q), f_\alpha(X) \rangle = \max_{X \in \text{CUT}_m^n} \langle f_\alpha(Q), X \rangle, \quad (3.14)$$

where the last inequality is due to the fact that  $\text{CUT}_m^n$  is closed under the action of  $f_\alpha$ . Likewise,  $\mathcal{E}_m^n$  is also closed under the action of  $f_\alpha$ , as shown in Section 3.2.1. Therefore, (3.14) also holds when replacing  $\text{CUT}_m^n$  by  $\mathcal{E}_m^n$ . Thus, by definition of the function  $\text{str}$ , see (3.8), the lemma follows.  $\square$

We provide an explicit example of such an ROC transformation. Let  $Q$  and  $X$  be Hermitian matrices of order  $n$ , with  $\text{diag}(Q) = \mathbf{0}$ . Then,

$$\langle Q, X \rangle = 2 \sum_{ij \in [n]^2, i < j} \text{Re}(\overline{Q_{ij}} X_{ij}). \quad (3.15)$$

Using  $((\alpha\alpha^H) \odot Q)_{ij} = Q_{ij}\alpha_i\overline{\alpha_j}$  for  $\alpha \in \mathcal{U}_m^n$ , it is easy to see how the Hadamard product transforms (3.15). However, we simplify notation by considering the vector  $[\alpha_0, \alpha_1, \dots, \alpha_{n-1}]^\top \in \mathcal{U}_m^n$ , and  $\beta = \alpha_0[1, \alpha_1, \dots, \alpha_{n-1}]^\top \in \mathcal{U}_m^n$ . Note that the first column of  $\beta\beta^H$  is given by  $[1 \ \alpha_1 \ \dots \ \alpha_{n-1}]^\top$ , so that

$$\frac{1}{2} \langle (\beta\beta^H) \odot Q, X \rangle = \text{Re} \left[ \sum_{j=2}^n \overline{Q_{1j}} \alpha_{j-1} X_{1j} + \sum_{ij \in [n]^2, 1 < i < j} \overline{Q_{ij}} \overline{\alpha_{i-1}} \alpha_{j-1} X_{ij} \right] \quad (3.16)$$

We exploit the equality above to derive the ROC equivalent inequalities in the next section. The following lemma shows that one can also consider the complex conjugate of matrix  $Q$ , and  $Q_\sigma$  as in (3.12), without changing the strength of the corresponding valid inequality. This results in *complex conjugate equivalent* and *permutation equivalent* inequalities respectively.

**Lemma 3.7.** *Let  $m, n \geq 2$  be integers and  $Q \in \mathcal{H}^n$ . For any permutation  $\sigma : [n] \rightarrow [n]$ , and  $Q_\sigma$  as in (3.12), we have that*

$$\text{str}(Q, m) = \text{str}(\overline{Q}, m) = \text{str}(Q_\sigma, m). \quad (3.17)$$

*Proof.* We prove only the first equality (3.17). The second equality in (3.17) can be proven in a similar manner. For  $Z \in \mathcal{H}^n$ , the complex conjugation function  $Z \rightarrow \overline{Z}$  is conjugate-linear, invertible, and satisfies an equation similar to (3.11), i.e.,  $\langle \overline{Z}, \overline{Z'} \rangle = \langle Z, Z' \rangle$ . Thus, (3.14) is also valid when  $f_\alpha$  is replaced with the complex conjugation function. Hence, the same arguments that prove Lemma 3.6 also the first equality in (3.17).  $\square$

We provide an example of inequalities that are equivalent under permutation symmetry of the corresponding matrix  $Q$ . Consider the inequalities on the right-hand side of (3.6). In particular, for fixed  $k \in [m]$  and distinct  $i, j, t, \ell \in [n]$ , consider the two inequalities

$$\operatorname{Re}(\bar{v}_k X_{ij}) \leq \cos\left(\frac{\pi}{m}\right), \quad \operatorname{Re}(\bar{v}_k X_{t\ell}) \leq \cos\left(\frac{\pi}{m}\right). \quad (3.18)$$

The two inequalities in (3.18) are equivalent under the permutation symmetry. Moreover, there is no combination of rotation symmetries and complex conjugation that transform the first inequality in (3.18) into the second.

**Example 3.8** (max-3-cut). The max-3-cut problem is to partition the vertex set of a graph into 3 disjoint subsets such that the total weight of edges joining different sets is maximized. This problem can be modeled using  $\text{CUT}_3^n$  as noted by Goemans and Williamson [118]. The same authors also derived a CSDP relaxation for the max-3-cut problem whose feasible set is  $\mathcal{E}_3^n$ , see (3.3).

To model the max-3-cut problem on some graph  $G = (V, E)$ ,  $|V| = n$ , we may associate to each vertex  $i \in V$  a variable  $x_i \in \mathcal{U}_3$ , see (3.1). The objective value of any variable assignment (i.e., cut) equals the number of edges  $\{i, j\} \in E$  for which  $x_i \neq x_j$ . Note that, if  $x_i \neq x_j$ , then  $\bar{x}_i x_j \in \mathcal{U}_3 \setminus \{1\}$ . Since  $\{\operatorname{Re}(z) : z \in \mathcal{U}_3 \setminus \{1\}\} = -1/2$ , we have

$$\frac{2}{3} \operatorname{Re}(1 - \bar{x}_i x_j) = \begin{cases} 1, & \text{if } x_i \neq x_j \\ 0, & \text{else.} \end{cases}$$

Thus, for a graph  $G$ , the objective value of the cut induced by  $x \in \mathcal{U}_3^n$  is given by

$$v(G, x) = \frac{2}{3} \sum_{\{i, j\} \in E} \operatorname{Re}(1 - \bar{x}_i x_j). \quad (3.19)$$

For the complete graph of order 4, denoted by  $K_4$ , it is not difficult to verify that  $v(K_4, x) \in \{0, 3, 4, 5\}$  for all  $x \in \mathcal{U}_3^4$ . That is, any 3-cut of  $K_4$  cuts either 0, 3, 4 or 5 edges. By rewriting (3.19) for  $G = K_4$ , we find that

$$\sum_{i < j} \operatorname{Re}(\bar{x}_i x_j) = 6 - \frac{3}{2} v(K_4, x) \in \left\{0, \pm \frac{3}{2}, 6\right\}.$$

Therefore, the inequality  $\operatorname{Re}(X_{ij} + X_{ik} + X_{il} + X_{jk} + X_{j\ell} + X_{\ell k}) \geq -3/2$  is valid for  $\text{CUT}_3^n$ , along with its ROC equivalent inequalities. We show in the next section that this inequality is not implied by  $\mathcal{E}_3^n$ , by proving that the strength, see (3.8), of the inequality is strictly greater than 1.  $\triangle$

### 3.2.3 Generalized complex triangle and quadrangle inequalities

In this section, we first generalize the gap inequalities [187] from  $\text{CUT}_2^n$  to  $\text{CUT}_m^n$ , with  $m > 2$  integer. Then, we derive some valid inequalities for  $\text{CUT}_m^n$  for different values

of  $m$  by exploiting (3.9), and compute their strength. In particular, we show that the generalized complex triangle and complex quadrangle inequalities may strengthen  $\text{CUT}_m^n$  for finite  $m \geq 2$ .

To derive the gap inequalities from [187], we set

$$\gamma_m(b) := \min_{x \in \mathcal{U}_m^n} |b^H x| \quad \text{and} \quad \sigma(b) := \sum_{i \in [n]} b_i, \quad (3.20)$$

for  $b \in \mathbb{C}^n$ , and  $B = bb^H - \text{Diag}(|b_1|^2, \dots, |b_n|^2)$ . The gap inequality, corresponding to some  $b \in \mathbb{R}^n$ , is then defined as

$$\langle B, X \rangle \geq 2 \sum_{1 \leq i < j \leq n} b_i b_j + \gamma_2(b)^2 - \sigma(b)^2 \quad \forall X \in \text{CUT}_2^n. \quad (3.21)$$

Note that Laurent and Poljak [187] define the gap inequality in terms of  $\{0, 1\}$  variables, rather than  $\{\pm 1\}$ , which explains the discrepancy between (3.21) and the gap inequality presented in [187]. We generalize (3.21) to  $\mathbb{C}$  in the following lemma.

**Lemma 3.9.** *Let  $b \in \mathbb{C}^n$ , and set  $B = bb^H - \text{Diag}(|b_1|^2, \dots, |b_n|^2)$ . Then, for  $\gamma_m$  and  $\sigma$  as in (3.20), we have that*

$$\min_{X \in \text{CUT}_m^n} \langle B, X \rangle = 2 \operatorname{Re} \left( \sum_{1 \leq i < j \leq n} b_i \bar{b}_j \right) + \gamma_m(b)^2 - \sigma(b) \overline{\sigma(b)}.$$

*Proof.* The result follows from the fact that

$$\gamma_m(b)^2 = \min_{X \in \text{CUT}_m^n} \langle bb^H, X \rangle = \min_{X \in \text{CUT}_m^n} \langle B, X \rangle + \|b\|^2,$$

and  $\|b\|^2 = \sigma(b) \overline{\sigma(b)} - 2 \operatorname{Re} \left( \sum_{1 \leq i < j \leq n} b_i \bar{b}_j \right)$ . □

We now consider the gap inequalities corresponding to  $b = \mathbf{1}_3$  and  $b = \mathbf{1}_4$ .

**Proposition 3.10.** *Let  $m \geq 2$ ,  $n \in \{3, 4\}$ ,  $Q_n = \mathbf{I}_n - \mathbf{J}_n$ . We have that*

$$\max_{X \in \mathcal{E}_m^n} \langle Q_n, X \rangle = n,$$

and

$$\begin{aligned} & \max_{X \in \text{CUT}_m^n} \langle Q_n, X \rangle \\ &= \begin{cases} -4 \cos \left( \frac{2 \lfloor m/3 \rfloor \pi}{m} \right) - 2 \cos \left( \frac{4 \lfloor m/3 \rfloor \pi}{m} \right) & \text{if } n = 3 \text{ and } \gcd(n, m) = 1, \\ -2 - 8 \cos \left( \frac{2 \lfloor m/2 \rfloor \pi}{m} \right) - 2 \cos \left( \frac{4 \lfloor m/2 \rfloor \pi}{m} \right) & \text{if } n = 4 \text{ and } \gcd(n, m) = 1, \\ n & \text{else.} \end{cases} \quad (3.22) \end{aligned}$$

*Proof.* For any  $Y \in \mathcal{E}_m^n$ , the value  $\langle Q_n, Y \rangle$  provides a lower bound on  $\max_{X \in \mathcal{E}_m^n} \langle Q_n, X \rangle$ . Specifically for  $Y = (n\mathbf{I}_n - \mathbf{J}_n)/(n-1)$ , we have  $\max_{X \in \mathcal{E}_m^n} \langle Q_n, X \rangle \geq \langle Q_n, Y \rangle = n$ .

Moreover, we have for all  $X \in \mathcal{E}_m^n$ ,  $\langle Q_n, X \rangle = \langle \mathbf{I}_n, X \rangle - \langle \mathbf{J}_n, X \rangle = n - \langle \mathbf{J}_n, X \rangle \leq n$ , since  $\mathbf{J}_n$  and  $X$  are PSD. Thus  $\max_{X \in \mathcal{E}_m^n} \langle Q_n, X \rangle = n$ .

For optimization over  $\text{CUT}_m^n$ , note that  $(-Q_n) = \mathbf{1}_n \mathbf{1}_n^H - \text{Diag}(\mathbf{1}_n)$ , and we may apply Lemma 3.9, for  $b = \mathbf{1}_n$ . Consequently,  $\sigma(\mathbf{1}_n) = n$ , and

$$\begin{aligned} \max_{X \in \text{CUT}_m^n} \langle Q_n, X \rangle &= - \min_{X \in \text{CUT}_m^n} \langle -Q_n, X \rangle \\ &= n^2 - 2 \binom{n}{2} - \gamma_m(\mathbf{1}_n)^2 = n - \gamma_m(\mathbf{1}_n)^2. \end{aligned} \quad (3.23)$$

It remains to determine  $\gamma_m(\mathbf{1}_n) = \min_{x \in \mathcal{U}_m^n} |\mathbf{1}^H x|$ . It is clear that when  $n = 3$  and  $m$  a multiple of 3, or  $n = 4$  and  $m$  even,  $\gamma_m(\mathbf{1}) = 0$  (since then there exist  $n$   $m$ th roots of unity that sum to 0).

For  $n = 3$  and  $m$  not a multiple of 3, geometric arguments from [230] show that the optimal value is attained for  $x^* = (1, z, \bar{z})^\top$ , where  $z = \exp\left(\frac{2\lfloor m/3 \rfloor \pi}{m} \mathbf{i}\right)$ . Then,

$$\begin{aligned} \gamma_m(\mathbf{1}_3)^2 &= |\mathbf{1}_3^H x^*|^2 = \left(1 + 2 \cos\left(\frac{2\lfloor m/3 \rfloor \pi}{m}\right)\right)^2 \\ &= 3 + 4 \cos\left(\frac{2\lfloor m/3 \rfloor \pi}{m}\right) + 2 \cos\left(\frac{4\lfloor m/3 \rfloor \pi}{m}\right), \end{aligned} \quad (3.24)$$

and the result follows from substituting (3.24) in (3.23).

For  $n = 4$  and  $m$  odd, similar geometric arguments from [230] show that the minimizer of  $\gamma_m(\mathbf{1}_4)$  is given by  $x^* = (1, 1, z, \bar{z})^\top$ , where  $z = \exp\left(\frac{2\lfloor m/2 \rfloor \pi}{m} \mathbf{i}\right)$ . Using this to compute  $\gamma_m(\mathbf{1}_4)^2$ , and substituting the result in (3.23) yields the proof.  $\square$

The coefficients of the valid inequalities (3.22) can be multiplied by elements from  $\mathcal{U}_m^n$  without altering their strength, see Lemma 3.6. Let us present these ROC equivalent inequalities explicitly in the following corollary.

**Corollary 3.11.** *Let  $m \geq 2$ ,  $n \in \{3, 4\}$ ,  $Q_n = \mathbf{I}_n - \mathbf{J}_n$ . For  $n = 3$ , the ROC equivalent inequalities of the inequality induced by Proposition 3.10 read*

$$-2 \operatorname{Re}(\alpha_1 X_{12} + \alpha_2 X_{13} + \bar{\alpha}_1 \alpha_2 X_{23}) \leq \max_{X \in \text{CUT}_m^3} \langle Q_3, X \rangle, \quad (3.25)$$

where  $\alpha \in \mathcal{U}_m^2$ . For  $n = 4$ , we have the following ROC equivalent inequalities

$$\begin{aligned} &-2 \operatorname{Re}(\alpha_1 X_{12} + \alpha_2 X_{13} + \alpha_3 X_{14} + \bar{\alpha}_1 \alpha_2 X_{23} + \bar{\alpha}_1 \alpha_3 X_{24} + \bar{\alpha}_2 \alpha_3 X_{34}) \\ &\leq \max_{X \in \text{CUT}_m^4} \langle Q_4, X \rangle, \end{aligned} \quad (3.26)$$

where  $\alpha \in \mathcal{U}_m^3$ . Lastly,  $\mathbf{str}(Q_n, m) > 1$ , see (3.8), if and only if  $\gcd(n, m) = 1$ .

*Proof.* The inequalities (3.25) and (3.26) are obtained from (3.9) and (3.16) where  $Q := \mathbf{I}_n - \mathbf{J}_n$ .

To show that  $\mathbf{str}$  is strictly greater than 1 whenever  $\gcd(n, m) = 1$ , we consider again separate cases. Let first  $n = 3$  and  $m \equiv 1 \pmod{3}$ , which implies that  $\lfloor m/3 \rfloor =$

$(m-1)/3$ . Substituting this in (3.22) for  $n = 3$ , and using that  $\cos(2z) = 2\cos^2(z) - 1$ , we find that

$$\max_{X \in \text{CUT}_m^3} \langle Q_3, X \rangle = 2 - 4\cos(z_m) - 4\cos^2(z_m) := g(m), \text{ for } z_m := \frac{2(m-1)\pi}{3m}$$

and  $m \equiv 1 \pmod{3}$ . Observe that  $g(m)$  is a concave quadratic function in  $\cos(z_m)$  that is maximized for

$$\cos(z_m) = -\frac{1}{2} \implies z_m = \frac{2\pi}{3} + 2k\pi \vee z_m = \frac{4\pi}{3} + 2k\pi, \quad k \in \mathbb{Z}.$$

The maximum of  $g(m)$  equals 3, which is not attained for any  $m \geq 1$ , since

$$m \geq 1 \implies 0 \leq z_m = \frac{2(m-1)\pi}{3m} < \frac{2\pi}{3}.$$

Hence, the maximum value of 3 is not attained for finite  $m \geq 1$  in case  $m \equiv 1 \pmod{3}$ . Thus,

$$m \equiv 1 \pmod{3} \implies \max_{X \in \text{CUT}_m^3} \langle Q_3, X \rangle < 3 = \max_{X \in \mathcal{E}_m^3} \langle Q_3, X \rangle. \quad (3.27)$$

The equality in (3.27) follows from Proposition 3.10. It follows from (3.27) that the strength of the corresponding inequality is strictly greater than 1. The proof for other values of  $n$  and  $m$  follows similarly.  $\square$

Thus, the inequalities given by Corollary 3.11 separate  $\mathcal{E}_m^n$  from  $\text{CUT}_m^n$  only when  $\gcd(n, m) = 1$ . The strength of these inequalities is greater for smaller values of  $m$ , as in the limit to infinity, the optimal value of the discrete programming problem in Proposition 3.10 equals  $n$ . For numerical evaluation of the strength of these inequalities, see Table 3.2 in Section 3.7.1. Note that the inequalities from Example 3.8 can be also derived from Proposition 3.10 for  $n = 4$  and  $m = 3$ .

Let us highlight Proposition 3.10 for the real case, i.e., for  $m = 2$ . Considering  $n = 3$ , the expressions in Proposition 3.10 then provide

$$\max_{X \in \text{CUT}_2^3} \langle Q_3, X \rangle = 2,$$

and since  $\mathcal{U}_2 = \{\pm 1\}$ , the inequalities (3.25) then reduce to the well-known triangle inequalities (3.13) (after appropriate scaling). Hence, the inequalities (3.25) may be considered as *generalized complex triangle inequalities*.

Similarly, the inequalities (3.26) for  $n = 4$  can be considered as *complex quadrangle inequalities*. For the real case,  $m = 2$ , we have that  $\gcd(n, m) = \gcd(4, 2) = 2 > 1$ . Therefore, by Corollary 3.11, the strength of these inequalities equals 1 for  $m = 2$ . Thus, the quadrangle inequalities are implied by  $\mathcal{E}_2^4$ . This clarifies why in the real case, the triangle, pentagonal, heptagonal (etc.) inequalities are well known, in contrast to real quadrangle inequalities. Note that real triangle, pentagonal, heptagonal, etc., inequalities belong to the family of hypermetric inequalities that are considered as a special case of the gap inequalities (3.21).

The inequalities derived in Corollary 3.11 are valid for  $n \in \{3, 4\}$ , and therefore can be applied to the principal submatrices of matrices in  $\mathcal{E}_m^{\tilde{n}}$  for  $\tilde{n} > 4$ . Thus, the complex triangle and quadrangle inequalities apply to all  $n$ . We present this idea more formally in the next section, see (3.36), and exploit it in Section 3.7.

In particular, the inequalities derived in Corollary 3.11 are also valid for  $n = m = 3$ , but already satisfied by matrices in  $\mathcal{E}_3^3$  (since their strength equals 1). This contrasts the real case, as the inequality from Proposition 3.10 for  $n = 3$  and  $m = 2$  is not satisfied by all matrices in  $\mathcal{E}_2^3$ . In the next section, we determine all facet defining inequalities of  $\text{CUT}_3^3$ , and show that some of them are violated by matrices in  $\mathcal{E}_3^3$ .

### 3.3 An exact description of $\text{CUT}_3^3$

We study  $\text{CUT}_3^3$  by studying the set

$$\mathcal{V}(\text{CUT}_3^3) := \left\{ x \in \mathbb{C}^3 : \begin{bmatrix} 1 & x_1 & x_2 \\ \overline{x_1} & 1 & x_3 \\ \overline{x_2} & \overline{x_3} & 1 \end{bmatrix} \in \text{CUT}_3^3 \right\}. \quad (3.28)$$

We define the sets  $\mathcal{V}(\text{CUT}_m^n)$ , in the remainder of the chapter, analogously. It is clear that there exists a bijection between the sets  $\mathcal{V}(\text{CUT}_3^3)$  and  $\text{CUT}_3^3$ . Since  $\mathcal{V}(\text{CUT}_3^3)$  is small, we can tractably compute its facets. We first require some intermediate results.

**Proposition 3.12.** *The inequality*

$$\text{Re} \left( \mathbf{i}x_1 + e^{\pi\mathbf{i}/6}x_2 + \mathbf{i}x_3 \right) \leq \frac{\sqrt{3}}{2}, \quad (3.29)$$

is facet defining for  $\mathcal{V}(\text{CUT}_3^3)$ . Additionally, the three linear inequalities that ensure  $x_i \in \text{Conv}(\mathcal{U}_3)$  for  $i \in [3]$ , see (3.6), are also facet defining.

*Proof.* We consider  $\mathcal{V}(\text{CUT}_3^3)$  as a real space of dimension 6. Consider the six vectors  $z_\theta := (e^{\theta_1\mathbf{i}}, e^{\theta_2\mathbf{i}}, e^{(\theta_2-\theta_1)\mathbf{i}})^\top$ , where  $\theta = (\theta_1, \theta_2)$  and

$$\theta \in \left\{ (0, 0), \left( \frac{2\pi}{3}, 0 \right), \left( \frac{4\pi}{3}, 0 \right), \left( 0, \frac{4\pi}{3} \right), \left( \frac{4\pi}{3}, \frac{2\pi}{3} \right), \left( \frac{4\pi}{3}, \frac{4\pi}{3} \right) \right\}.$$

These six vectors satisfy the following properties: each  $z_\theta$  corresponds to the upper triangular entries of

$$Z_\theta := \begin{bmatrix} 1 \\ e^{-\theta_1\mathbf{i}} \\ e^{-\theta_2\mathbf{i}} \end{bmatrix} \begin{bmatrix} 1 \\ e^{-\theta_1\mathbf{i}} \\ e^{-\theta_2\mathbf{i}} \end{bmatrix}^H \in \text{CUT}_3^3,$$

as in (3.28), and therefore,  $z_\theta \in \mathcal{V}(\text{CUT}_3^3)$ . Moreover, since  $Z_\theta$  is an extreme point of  $\text{CUT}_3^3$ ,  $z_\theta$  is an extreme point of  $\mathcal{V}(\text{CUT}_3^3)$ . Additionally, all the six vectors  $z_\theta$  satisfy (3.29) with equality.

Let us now consider the six real vectors  $\tilde{z}_\theta := [\text{Re}(z_\theta)^\top \quad \text{Im}(z_\theta)^\top]^\top \in \mathbb{R}^6$ . It is not difficult to verify that the  $\tilde{z}_\theta$  are affinely independent. This fact, together with the fact that all extreme points of  $\mathcal{V}(\text{CUT}_3^3)$  satisfy the inequality (3.29), implies that (3.29) is facet defining. The proof that  $x_i \in \text{Conv}(\mathcal{U}_3)$  induces three facets follows similarly.  $\square$

**Remark 3.13.** Using similar arguments, one can also show that the cut from Proposition 3.10, for  $n = 3$  and  $m = 4$ , is facet defining for  $\mathcal{V}(\text{CUT}_4^3)$ .  $\triangle$

We also compute the strength, see (3.8), of the inequality (3.29). To do so, note that the unique Hermitian matrix corresponding to (3.29) is given by

$$Q := \frac{1}{2} \begin{bmatrix} 0 & \mathbf{i} & e^{\pi\mathbf{i}/6} \\ -\mathbf{i} & 0 & \mathbf{i} \\ e^{-\pi\mathbf{i}/6} & -\mathbf{i} & 0 \end{bmatrix}. \quad (3.30)$$

That is, inequality (3.29) is equivalent to  $\langle Q, X \rangle \leq \sqrt{3}/2$  for  $X \in \text{CUT}_3^3$ .

**Lemma 3.14.** *For  $Q$  as in (3.30), we have that*

$$\text{str}(Q, 3) = \frac{\sqrt{3} \cos\left(\frac{\pi}{18}\right)}{\cos\left(\frac{\pi}{9}\right)} \approx 1.81521.$$

*Proof.* See Page 204 in Appendix B.2.  $\square$

**Remark 3.15.** Inequality (3.29) is not a gap inequality, see Lemma 3.9. Indeed, (3.29) is a gap inequality if and only if its corresponding matrix  $Q$ , see (3.30), satisfies  $Q = bb^H - \text{Diag}(|b_1|^2, |b_2|^2, |b_3|^2)$  for some  $b \in \mathbb{C}^3$ . This equality implies that

$$b_1 \bar{b}_2 = b_2 \bar{b}_3 = \frac{1}{2} \mathbf{i} \text{ and } b_1 \bar{b}_3 = \frac{1}{2} e^{\pi\mathbf{i}/6},$$

which is inconsistent: note that  $b_1 |b_2|^2 \bar{b}_3 = (\mathbf{i}/2)^2 = -1/4$ , which implies that  $b_1 \bar{b}_3 \in \mathbb{R}$ . This contradicts the fact that  $b_1 \bar{b}_3 = e^{\pi\mathbf{i}/6}/2 \notin \mathbb{R}$ .  $\triangle$

**Lemma 3.16.** *The ROC equivalent inequalities (see Lemma 3.6) and the complex conjugate equivalent inequalities (see Lemma 3.7) of facet defining inequalities of  $\mathcal{V}(\text{CUT}_m^n)$ , are again facet defining.*

*Proof.* Let  $g(x) \leq c$ ,  $c \in \mathbb{R}$ , be a facet defining inequality for  $\mathcal{V}(\text{CUT}_m^n) \subseteq \mathbb{C}^{\binom{n}{2}}$ . Then there exist vectors  $y^j \in \mathcal{V}(\text{CUT}_m^n)$ ,  $j \in [2^{\binom{n}{2}}]$ , that satisfy  $g(y^j) = c$  and are affinely independent over the reals. That is,

$$\begin{bmatrix} 1 & 1 & \cdots & 1 \\ y^1 & y^2 & \cdots & y^{2^{\binom{n}{2}}} \end{bmatrix} v = \mathbf{0}, v \in \mathbb{R}^{2n} \iff v = \mathbf{0}. \quad (3.31)$$

Additionally, for each such  $y^j$ , there exists a  $Y^j \in \text{CUT}_m^n$  such that the vector  $y^j$  corresponds to the upper triangular entries of  $Y^j$ . Let us slightly abuse the notation of (3.28), and write this relation as  $\mathcal{V}(Y^j) = y^j$ , where the linear function  $\mathcal{V} : \text{CUT}_m^n \rightarrow$

$\mathbb{C}^{\binom{n}{2}}$  returns the upper triangular entries of its input matrix. We can write  $g(x) \leq c$  in terms of matrices as  $\langle G, X \rangle \leq c$  for some  $G \in \mathcal{H}^n$ , and  $X \in \text{CUT}_m^n$ . By construction,

$$\langle G, Y^j \rangle = c. \quad (3.32)$$

Recall now the symmetries of  $\text{CUT}_m^n$  (and therefore also  $\mathcal{V}(\text{CUT}_m^n)$ ), as outlined in Section 3.2.1. In particular, recall the group action  $f_\alpha(Z) = (\alpha\alpha^H) \odot Z$  for some  $\alpha \in \mathcal{U}_m^n$ . Denote by  $\tilde{g}(x) \leq c$  the inequality that is ROC equivalent with  $g(x) \leq c$ , following a rotation with some  $\alpha \in \mathcal{U}_m^n$ . This inequality may be written as  $\langle f_\alpha(G), X \rangle \leq c$  for  $X \in \text{CUT}_m^n$ . From (3.11) and (3.32), we have  $\langle f_\alpha(G), f_\alpha(Y^j) \rangle = c$ .

Therefore, the vectors  $\tilde{y}^j := \mathcal{V}(f_\alpha(Y^j)) \in \mathcal{V}(\text{CUT}_m^n)$  satisfy  $\tilde{g}(\tilde{y}^j) = c$ . Note that

$$\tilde{y}^j = \text{Diag}(\alpha_1\bar{\alpha}_2, \alpha_1\bar{\alpha}_3, \dots, \alpha_{n-1}\bar{\alpha}_n)y^j. \quad (3.33)$$

Using (3.31) and (3.33), it follows that the vectors  $\tilde{y}^j$  are also affinely independent, since

$$\begin{bmatrix} 1 & \dots & 1 \\ \tilde{y}^1 & \dots & \tilde{y}^{2\binom{n}{2}} \end{bmatrix} v = \text{Diag}(1, \alpha_1\bar{\alpha}_2, \alpha_1\bar{\alpha}_3, \dots, \alpha_{n-1}\bar{\alpha}_n) \begin{bmatrix} 1 & \dots & 1 \\ y^1 & \dots & y^{2\binom{n}{2}} \end{bmatrix} v = \mathbf{0},$$

if and only if  $v = \mathbf{0}$ . Hence, the result follows. The proof for complex conjugate equivalent inequalities is similar.  $\square$

Let  $F$  denote the number of facets of  $\mathcal{V}(\text{CUT}_3^3)$ . Note that the facet defining inequality (3.29) has 9 ROC equivalent inequalities (counting itself), see (3.16), and its complex conjugate equivalent inequality also has 9 ROC equivalent inequalities (counting itself). Moreover, the three linear inequalities that ensure  $x_i \in \text{Conv}(\mathcal{U}_3)$  for  $i \in [3]$  are also facet defining (and ROC equivalent). Thus,

$$F \geq 18 + 9 = 27. \quad (3.34)$$

We are now ready to show that these 27 inequalities fully describe the set  $\mathcal{V}(\text{CUT}_3^3)$ .

**Theorem 3.17.** *The set  $\mathcal{V}(\text{CUT}_3^3)$  admits the following linear description:*

$$\mathcal{V}(\text{CUT}_3^3) = \left\{ x \in \mathbb{C}^3 : \begin{array}{l} x \in \text{Conv}(\mathcal{U}_3^3), \text{Re}(\eta x) \leq \frac{\sqrt{3}}{2}, \text{Re}(\bar{\eta}x) \leq \frac{\sqrt{3}}{2}, \\ \eta = (\alpha_1\mathbf{i}, \alpha_2e^{\pi\mathbf{i}/6}, \bar{\alpha}_1\alpha_2\mathbf{i}), \alpha \in \mathcal{U}_3^2. \end{array} \right\}. \quad (3.35)$$

*Proof.* The Upper-bound theorem for convex polytopes [223] states the following: for any convex  $d$ -dimensional polytope  $P$  with  $v$  vertices, the number of  $j$ -dimensional faces (see Definition B.3 on Page 205 in Appendix B.2) is upper bounded by some explicit number  $f_j(v, d)$ . For our purposes, we consider  $\mathcal{V}(\text{CUT}_3^3)$  as 6-dimensional real polytope. As its facets are 5-dimensional faces, the number of facets  $F$  satisfies  $F \leq f_5(9, 6) = 30$ , see, e.g., [101, Sect. 1, Thm. 4]. Combined with (3.34), this implies  $27 \leq F \leq 30$ . We prove now, by contradiction, that  $F = 27$ . Thus, assume that  $27 < F \leq 30$ . If that is the case, then there must exist some facet defining inequality  $\text{Re}(\beta_1x_1 + \beta_2x_2 + \beta_3x_3) \leq c$ , which is missing from the right hand side of (3.35). Note

that the vector  $\beta \in \mathbb{C}^3$  contains at least two nonzero entries: if  $\beta$  were to contain only a single nonzero entry, the inequality concerns only a single variable, say  $x_1$ . But the restriction  $x_1 \in \text{Conv}(\mathcal{U}_3)$  is already included in (3.35), and clearly cannot be made tighter.

Thus,  $\beta$  contains two or three nonzero entries. Now there must exist at least 8 other ROC equivalent inequalities, that are also facet defining. This contradicts the result  $F \leq 30$ , which completes the proof.  $\square$

On Page 201 in Appendix A.6, we verify Proposition 3.12 and Theorem 3.17 by listing all 27 facet defining inequalities found using the SageMath software [295].

We refer to the inequalities in (3.35), induced by  $\eta$ , as *the triangle facets (of  $\text{CUT}_3^3$ )*. One can strengthen the CSDP relaxation CSDP-P on Page 59 by adding the triangle facets to the complex ellipsope  $\mathcal{E}_3^n$ . Let us denote the resulting feasible set by:

$$\mathbf{T}(\mathcal{E}_3^n) = \{X \in \mathcal{E}_3^n : X_J \in \text{CUT}_3^3, \forall J \subseteq [n], |J| = 3\}. \quad (3.36)$$

Here,  $X_J$  denotes the  $|J| \times |J|$  principal submatrix of  $X$ , with rows and columns indicated by  $J$ .

### 3.4 Efficiently reformulating a class of CSDPs

It is well known that the max-3-cut problem can be modeled using  $\text{CUT}_3^n$ , as demonstrated in Example 3.8, and first shown by Goemans and Williamson [118]. To approximate the max-3-cut problem, one can solve a CSDP over  $\mathcal{E}_3^n$ . On the other hand, Frieze and Jerrum [95] approximate the max-3-cut problem by a real SDP having matrix variables of order  $n$ , that is equivalent to the CSDP over  $\mathcal{E}_3^n$ . Here we specify a class of CSDPs that can be efficiently reformulated as real SDPs.

Modern SDP solvers solve CSDPs by representing  $n \times n$  Hermitian matrices as  $2n \times 2n$  symmetric matrices, via

$$X \in \mathcal{H}_+^n \iff \tilde{X} = \begin{bmatrix} \text{Re}(X) & \text{Im}(X) \\ -\text{Im}(X) & \text{Re}(X) \end{bmatrix} \in \mathcal{S}_+^{2n}, \quad (3.37)$$

see also [114]. Consequently, CSDPs with matrix order  $n$ , require doubling the order to  $2n$ , and are therefore computationally more challenging to solve than real SDPs with matrix order  $n$ .

**Remark 3.18.** To the best of our knowledge, the only SDP solvers that handle CSDPs directly, rather than reformulating them to equivalent real SDPs, are SeDuMi [289], SDOLab [113] and Hypatia [59]. Despite this theoretical advantage, we observed that these solvers were still slower than MOSEK [228] for solving CSDPs. This was also observed in [304].  $\triangle$

Wang and Magron [306] introduce a real moment-Hermitian-SOS hierarchy for complex polynomial optimization problems with real coefficients, without doubling the order. Moreover, Wang and Magron show that their real hierarchy is equivalent to the complex hierarchy from the literature. Here, we provide another class of problems for which a CSDP can be equivalently formulated as a real SDP of the same size.

**Proposition 3.19.** *Let  $U \subseteq \mathcal{H}_+^n$  be a non-empty compact convex set, and  $W \in \mathcal{S}^n$ . Then*

$$\max_{X \in U} \langle W, X \rangle = \max_{X \in \text{Re}(U)} \langle W, X \rangle.$$

*If in addition  $\text{Re}(U) \subseteq U$ , then  $\arg \max_{X \in \text{Re}(U)} \langle W, X \rangle \subseteq \arg \max_{X \in U} \langle W, X \rangle$ .*

*Proof.* Since  $W$  is a real matrix,  $\langle W, X \rangle = \langle W, \text{Re}(X) \rangle$ . Therefore,  $\max_{X \in U} \langle W, X \rangle = \max_{X \in U} \langle W, \text{Re}(X) \rangle = \max_{X \in \text{Re}(U)} \langle W, X \rangle$ . The inclusion  $\arg \max_{X \in \text{Re}(U)} \langle W, X \rangle \subseteq \arg \max_{X \in U} \langle W, X \rangle$  follows trivially from  $\text{Re}(U) \subseteq U$ .  $\square$

A sufficient condition for  $\text{Re}(U) \subseteq U$  to hold, is that  $U$  is closed under complex conjugation and convex. Indeed, if  $U$  is closed under complex conjugation, then  $X \in U \iff \bar{X} \in U$ , and by convexity of  $U$ ,  $\text{Re}(X) = (X + \bar{X})/2 \in U$ . Note that the sets  $\mathcal{E}_m^n$ , see (3.3), are closed under complex conjugation and convex.

### 3.4.1 The case max-3-cut

We investigate the implications of Proposition 3.19, for the case that the underlying (C)SDP corresponds to a max-3-cut relaxation, see Example 3.8.

Let us first formulate the max-3-cut problem as a real program. Without loss of generality, we assume that the given graph is the complete graph on  $n$  vertices, with edge weights  $w_{ij} \in \mathbb{R}$ ,  $i, j \in [n]$ ,  $i < j$ . Following [95], let  $\mathbf{a}^1$ ,  $\mathbf{a}^2$  and  $\mathbf{a}^3$  be a set of unit length vectors in  $\mathbb{R}^3$  satisfying

$$(\mathbf{a}^i)^\top \mathbf{a}^j = \begin{cases} 1, & \text{if } i = j, \\ -\frac{1}{2}, & \text{else.} \end{cases} \quad (3.38)$$

Frieze and Jerrum [95] model the max-3-cut problem as

$$\max_y \frac{2}{3} \sum_{i < j} w_{ij} (1 - y_i^\top y_j) \quad \text{s.t.} \quad y_i \in \{\mathbf{a}^1, \mathbf{a}^2, \mathbf{a}^3\} \quad \forall i \in [n].$$

We investigate the feasible set of this program in terms of matrices. This set is given by

$$\begin{aligned} \text{Re}(\text{CUT}_3^n) &:= \{\text{Re}(Y) : Y \in \text{CUT}_3^n\} \\ &= \text{Conv} \{Y \in \mathcal{S}_+^n : \exists y_1, \dots, y_n \in \{\mathbf{a}^1, \mathbf{a}^2, \mathbf{a}^3\} \text{ s.t. } Y_{ij} = y_i^\top y_j \quad \forall i, j \in [n]\}. \end{aligned}$$

To understand the second equality above, note that the objective in the Frieze and Jerrum model and (3.19) are similar. That is,  $\text{Re}(\bar{x}_i x_j)$  is equal to the right-hand side of (3.38), for  $x_i, x_j \in \mathcal{U}_3$ .

As  $\text{CUT}_3^n$  satisfies the conditions for  $U$  in Proposition 3.19, we have, for  $W \in \mathcal{S}^n$ ,

$$\max_{X \in \text{CUT}_3^n} \langle W, X \rangle = \max_{Y \in \text{Re}(\text{CUT}_3^n)} \langle W, Y \rangle.$$

Thus,  $\text{Re}(\text{CUT}_3^n)$  is strictly smaller than  $\text{CUT}_3^n$ , but attains the same maxima of real linear forms, which we maximize for the max-3-cut problem. The same relation holds for the sets

$$\text{Re}(\mathcal{E}_3^n) := \{\text{Re}(X) : X \in \mathcal{E}_3^n\} = \left\{ X \in \mathcal{S}_+^n : \text{diag}(X) = \mathbf{1}_n, X_{ij} \geq -\frac{1}{2}, \forall i, j \in [n] \right\},$$

and  $\mathcal{E}_3^n$ , as it was already observed by Goemans and Williamson [118]. Note that  $\text{Re}(\mathcal{E}_3^n)$  corresponds to the feasible set of the max-3-cut SDP relaxation by Frieze and Jerrum [95]. However, if the objective matrix  $W$  satisfies  $\text{Im}(W) \neq \mathbf{0}$ , then the CSDP cannot be reformulated to a real SDP with same size.

Let us now study a relation between  $\mathbf{T}(\mathcal{E}_3^n)$ , see (3.36), and

$$\text{Re}(\mathbf{T}(\mathcal{E}_3^n)) := \{\text{Re}(X) : X \in \mathbf{T}(\mathcal{E}_3^n)\}.$$

To do so, we determine the facets of  $\text{Re}(\mathcal{V}(\text{CUT}_3^3))$  in the following lemma.

**Lemma 3.20.** *The set  $\text{Re}(\mathcal{V}(\text{CUT}_3^3)) := \{\text{Re}(x) : x \in \mathcal{V}(\text{CUT}_3^3)\}$ , see (3.35), is given by*

$$\text{Re}(\mathcal{V}(\text{CUT}_3^3)) = \left\{ x \in \mathbb{R}^3 : \begin{array}{l} x_i \geq -\frac{1}{2} \quad \forall i \in [3], x_1 + x_2 - x_3 \leq 1, \\ x_1 - x_2 + x_3 \leq 1, -x_1 + x_2 + x_3 \leq 1 \end{array} \right\}. \quad (3.39)$$

*Proof.* Starting from (3.35), we consider the following three vectors:  $\eta = (\mathbf{i}, e^{\pi\mathbf{i}/6}, \mathbf{i})$ ,  $\eta_1 = (e^{4\pi\mathbf{i}/3}\mathbf{i}, e^{\pi\mathbf{i}/6}, e^{-4\pi\mathbf{i}/3}\mathbf{i})$  and  $\eta_2 = (-e^{2\pi\mathbf{i}/3}\mathbf{i}, e^{-\pi\mathbf{i}/6}, -e^{-2\pi\mathbf{i}/3}\mathbf{i})$ . Note that  $\eta_1$  can be obtained from  $\eta$  by performing a rotation of coefficients with  $(\alpha_1, \alpha_2) = (\exp(4\pi\mathbf{i}/3), 1)$ . Similarly,  $\eta_2$  can be obtained by taking  $\bar{\eta}$ , and then performing the rotation of coefficients with  $(\alpha_1, \alpha_2) = (\exp(2\pi\mathbf{i}/3), 1)$ .

Thus  $\text{Re}(\eta_1 x) \leq \sqrt{3}/2$ , and  $\text{Re}(\eta_2 x) \leq \sqrt{3}/2$  are both valid inequalities for  $\mathcal{V}(\text{CUT}_3^3)$ , see Section 3.2.2. Consequently, also the sum of these inequalities is valid for  $\mathcal{V}(\text{CUT}_3^3)$ . That is,

$$\text{Re}((\eta_1 + \eta_2)x) = \text{Re}\left(\sqrt{3}(x_1 + x_2 - x_3)\right) \leq \sqrt{3} \iff \text{Re}(x_1 + x_2 - x_3) \leq 1, \quad (3.40)$$

which corresponds to one of the inequalities given in (3.39). Inequality (3.40) describes a facet of  $\text{Re}(\mathcal{V}(\text{CUT}_3^3))$ , since the vectors  $[1, 1, 1]^\top$ ,  $[1, -\frac{1}{2}, -\frac{1}{2}]^\top$ , and  $[-\frac{1}{2}, 1, -\frac{1}{2}]^\top$  are affinely independent, contained in  $\text{Re}(\mathcal{V}(\text{CUT}_3^3))$ , and satisfy (3.40) with equality. The other inequalities in (3.39) can be found in a similar manner.

Lastly, it can be shown that (3.39) contains all facet defining inequalities via a similar argument as the one used in the proof of Theorem 3.17 on Page 70.  $\square$

The facets provided in Lemma 3.20 are also given in [55, Equation 1.3] (they are stated in terms of  $\{0, 1\}$  variables rather than  $\{-\frac{1}{2}, 1\}$  as in (3.38)). However, our derivation from complex space is new. Using facets from (3.39), one can optimize over  $\text{Re}(\mathbf{T}(\mathcal{E}_3^n))$ . Note also that  $\mathbf{T}(\mathcal{E}_3^n)$  satisfies the conditions for  $U$  in Proposition 3.19. Hence, it is beneficial to optimize over  $\text{Re}(\mathbf{T}(\mathcal{E}_3^n))$  instead of  $\mathbf{T}(\mathcal{E}_3^n)$  if the matrix  $W$  is real.

Table 3.1 investigates the difference in solving times for optimization over the feasible sets  $\text{Re}(\mathbf{T}(\mathcal{E}_3^n))$  and  $\mathbf{T}(\mathcal{E}_3^n)$ . For various values of  $n$ , we generate uniformly at random a real symmetric matrix  $C \in \{-5, -4, \dots, 4, 5\}^{n \times n}$ , and solve the problem of maximizing  $\langle C, X \rangle$  over  $X \in \text{Re}(\mathbf{T}(\mathcal{E}_3^n))$ , and over  $X \in \mathbf{T}(\mathcal{E}_3^n)$ . This maximization is repeated 5 times per value of  $n$ , and the average running time of those 5 runs is reported in Table 3.1. As solver, we used MOSEK [228]. Note that optimization over  $\text{Re}(\mathbf{T}(\mathcal{E}_3^n))$  and  $\mathbf{T}(\mathcal{E}_3^n)$  returns the same objective value since  $C$  is real, see Proposition 3.19. Table 3.1 clearly demonstrates that optimization over  $\text{Re}(\mathbf{T}(\mathcal{E}_3^n))$  is more efficient compared to optimization over  $\mathbf{T}(\mathcal{E}_3^n)$ . The first reason for this is that solving real SDPs is computationally cheaper than solving CSDPs, see (3.37). The other reason is that  $\text{Re}(\mathbf{T}(\mathcal{E}_3^n))$  contains less inequalities than  $\mathbf{T}(\mathcal{E}_3^n)$ ; compare (3.35) with (3.39).

The CSDP reformulation approach by Wang and Magron [306] mentioned in Section 3.4.1 does not apply to CSDPs over  $\mathbf{T}(\mathcal{E}_3^n)$ , as the facets provided in (3.35) have non-real coefficients. Our generalization, Proposition 3.19, shows that a real reformulation of same size is possible when only the objective is real, and the feasible set is closed under complex conjugation (as is the case for  $\mathbf{T}(\mathcal{E}_3^n)$ ).

$n$	Time (seconds)	
	$\mathbf{T}(\mathcal{E}_3^n)$	$\text{Re}(\mathbf{T}(\mathcal{E}_3^n))$
20	0.22	0.03
30	0.80	0.10
40	3.02	0.33
50	6.98	0.84
60	15.22	1.83
70	33.17	3.67
80	59.34	6.73
90	114.07	12.73
100	199.72	22.74

Table 3.1: Comparison of solving times of optimization over  $\mathbf{T}(\mathcal{E}_3^n)$  and  $\text{Re}(\mathbf{T}(\mathcal{E}_3^n))$ .

### 3.5 Second semidefinite lifting of $\text{CUT}_\infty^n$

In this section we study approximations of  $\text{CUT}_\infty^n$ , see (3.4). The approximation of  $\text{CUT}_\infty^4$  obtained from the second semidefinite lifting as proposed by Jarre et al. [150] is denoted here by  $\mathbf{L}(\mathcal{B}_2)$ . The matrices in the set  $\mathbf{L}(\mathcal{B}_2)$  are obtained as projections of certain Hermitian PSD matrices of order seven. We propose an approximation of  $\text{CUT}_\infty^4$  denoted by  $\mathbf{L}(\mathcal{B}_1)$ , whose elements are the projections of certain Hermitian PSD matrices of order six. Despite this difference in size of the lifted space, we show that  $\mathbf{L}(\mathcal{B}_1) = \mathbf{L}(\mathcal{B}_2)$  (Lemma 3.25). Additionally, we show that  $\mathbf{L}(\mathcal{B}_1)$  is also equivalent to the second semidefinite lifting of the complex Lasserre hierarchy proposed in [154] (Theorem 3.26), whose elements are the projections of certain Hermitian PSD matrices of order ten.

The results from this section imply that one may appropriately decrease the size of matrices in a CSDP relaxation of  $\text{CUT}_\infty^n$ , while preserving the strength of the relaxation, see Lemma 3.41. We also show that  $\mathbf{L}(\mathcal{B}_1)$  excludes all the rank 2 extreme points of  $\mathcal{E}_\infty^4$  (Theorem 3.31). Lastly, we show that all elements of  $\mathbf{L}(\mathcal{B}_1)$  satisfy a valid inequality for  $\text{CUT}_\infty^4$ , derived in [150] (Lemma 3.38).

We begin our analysis with the following result on the rank of extreme points of  $\mathcal{E}_\infty^n = \{X \in \mathcal{H}_+^n : \text{diag}(X) = \mathbf{1}_n\}$ . The extreme points of  $\mathcal{E}_\infty^n$  have been widely studied, see e.g., [56, 126, 194, 201].

**Lemma 3.21** ([201]). *The extreme points of  $\mathcal{E}_\infty^n$  have rank at most  $\sqrt{n}$ . Moreover, for every positive integer  $r \leq \sqrt{n}$ , the set  $\mathcal{E}_\infty^n$  contains rank  $r$  extreme points.*

In case  $n \leq 3$ , the extreme points of  $\mathcal{E}_\infty^n$  have rank 1, and thus  $\mathcal{E}_\infty^n = \text{CUT}_\infty^n$  for  $n \leq 3$ . Therefore, in the sequel, we consider the smallest non-trivial case, that is  $n = 4$ . In this case,  $\mathcal{E}_\infty^4$  contains rank 2 extreme points (see (3.58) on Page 80), unlike  $\text{CUT}_\infty^4$ , which shows that  $\text{CUT}_\infty^4$  is strictly contained in  $\mathcal{E}_\infty^4$ . This motivates the authors of [150] to investigate a second semidefinite lifting approximation to  $\text{CUT}_\infty^4$ . To present their lifting, we first require some notation and definitions.

For some  $p \in \mathbb{N}$ , let  $\mathcal{B} \subseteq \mathbb{Z}^p$  be a finite set satisfying  $\mathbf{0}_p \in \mathcal{B}$ . Consider a complex (truncated pseudo-moment) sequence  $(y_\alpha)_{\alpha \in \mathcal{B} - \mathcal{B}}$ , satisfying  $y_{\mathbf{0}} = 1$  and  $y_\alpha = \bar{y}_{-\alpha}$ , where  $\mathcal{B} - \mathcal{B} := \{\alpha - \beta : \alpha, \beta \in \mathcal{B}\}$ . We define the *complex moment matrix*  $M_{\mathcal{B}}(y)$ , indexed by the elements of  $\mathcal{B}$ , as the matrix

$$M_{\mathcal{B}}(y) := (y_{\alpha - \beta})_{\alpha, \beta \in \mathcal{B}}. \quad (3.41)$$

By the properties of  $y$ ,  $M_{\mathcal{B}}(y) \in \mathcal{H}^{|\mathcal{B}|}$  and  $\text{diag}(M_{\mathcal{B}}(y)) = \mathbf{1}_{|\mathcal{B}|}$ . Let  $\tilde{\mathbb{C}}[x]$  be the set of polynomials defined by

$$\tilde{\mathbb{C}}[x] := \left\{ \sum_{\alpha \in \mathbb{Z}^p} f_\alpha x^\alpha : f_\alpha \in \mathbb{C} \forall \alpha \in \mathbb{Z}^p \right\} \quad (3.42)$$

for

$$x^\alpha := \prod_{i \in [p]} x_i^{\alpha_i} \text{ where } x_i^{\alpha_i} = \begin{cases} x_i^{\alpha_i} & \text{if } \alpha_i \geq 0, \\ (\bar{x}_i)^{-\alpha_i} & \text{if } \alpha_i < 0. \end{cases}$$

Note that  $\text{Re}(x_i) = (x_i + \bar{x}_i)/2 \in \tilde{\mathbb{C}}[x]$ . We set

$$\mathcal{F}(\mathcal{B}) := \{M_{\mathcal{B}}(y) : y_{\mathbf{0}} = 1 \text{ and } y_\alpha = \bar{y}_{-\alpha}\} \cap \mathcal{H}_+^{|\mathcal{B}|}. \quad (3.43)$$

In this section, we study the sets

$$\mathbf{L}(\mathcal{B}_i) := \{X \in \mathcal{E}_\infty^4 : \exists Z \in \mathcal{F}(\mathcal{B}_i) \text{ satisfying } Z_{1:4,1:4} = X\} \quad i \in [6], \quad (3.44)$$

which are defined in terms of the (ordered) bases

$$\mathcal{B}_1 := \left\{ \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \begin{bmatrix} -1 \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix} \right\}, \mathcal{B}_2 := \mathcal{B}_1 \cup \left\{ \begin{bmatrix} 0 \\ -1 \\ 1 \end{bmatrix} \right\}, \quad (3.45)$$

and  $\mathcal{B}_3$  up to  $\mathcal{B}_6$ , which will be given later.

Observe that  $\mathcal{B}_1$  and  $\mathcal{B}_2$  do not contain monomial squares, i.e., for  $k \in \{1, 2\}$ , we have that  $\alpha \in \mathcal{B}_k \implies |\alpha_i| < 2$  for all  $i \in [3]$ . Theorem 3.26 shows that adding monomial squares to  $\mathcal{B}_1$  does not lead to a tighter approximation of  $\text{CUT}_\infty^4$ . A similar result follows for  $\mathcal{B}_2$ , see Corollary 3.27. An example that will be used throughout is the following:  $(M_{\mathcal{B}_2}(\mathbf{y}))_{\alpha, \beta} = L_{\mathbf{y}}(X_{\alpha, \beta})$ , for

$$X = \begin{bmatrix} 1 & \overline{x_1} & \overline{x_2} & \overline{x_3} & x_1 \overline{x_2} & x_1 \overline{x_3} & x_2 \overline{x_3} \\ x_1 & 1 & x_1 \overline{x_2} & x_1 \overline{x_3} & x_1^2 \overline{x_2} & x_1^2 \overline{x_3} & x_1 x_2 \overline{x_3} \\ x_2 & \overline{x_1} x_2 & 1 & x_2 \overline{x_3} & x_1 & x_1 x_2 \overline{x_3} & x_2^2 \overline{x_3} \\ x_3 & \overline{x_1} x_3 & \overline{x_2} x_3 & 1 & x_1 \overline{x_2} x_3 & x_1 & x_2 \\ \overline{x_1} x_2 & \overline{x_1^2} x_2 & \overline{x_1} & \overline{x_1} x_2 \overline{x_3} & 1 & x_2 \overline{x_3} & \overline{x_1} x_2^2 \overline{x_3} \\ \overline{x_1} x_3 & \overline{x_1^2} x_3 & \overline{x_1} x_2 \overline{x_3} & \overline{x_1} & \overline{x_2} x_3 & 1 & \overline{x_1} x_2 \\ \overline{x_2} x_3 & \overline{x_1} x_2 \overline{x_3} & \overline{x_2^2} x_3 & \overline{x_2} & x_1 \overline{x_2^2} x_3 & x_1 \overline{x_2} & 1 \end{bmatrix}, \quad (3.46)$$

where  $L_{\mathbf{y}} : \widetilde{\mathbb{C}}[x] \rightarrow \mathbb{C}$  is the linear *Riesz functional*, defined by

$$L_{\mathbf{y}}(f) = \sum_{\alpha \in \mathbb{Z}^p} f_{\alpha} y_{\alpha}, \quad (3.47)$$

see (3.42). Observe also that  $M_{\mathcal{B}_1}(\mathbf{y})$  is the upper left  $6 \times 6$  block of  $M_{\mathcal{B}_2}(\mathbf{y})$ .

We refer to the sets  $\mathbf{L}(\mathcal{B}_i)$  as semidefinite liftings of  $\text{CUT}_\infty^4$ , since

$$\text{CUT}_\infty^4 \subseteq \mathbf{L}(\mathcal{B}_2) \subseteq \mathbf{L}(\mathcal{B}_1) \subseteq \mathcal{E}_\infty^4. \quad (3.48)$$

Jarre et al. [150] propose  $\mathbf{L}(\mathcal{B}_2)$  as a tighter approximation of  $\text{CUT}_\infty^4$  than  $\mathcal{E}_\infty^4$ .

**Remark 3.22.** Jarre et al. originally present their relaxation as  $\mathbf{L}(\mathcal{B}_3)$ , see (3.44), for

$$\mathcal{B}_3 := \left\{ \mathbf{0}_4, \begin{bmatrix} 1 \\ -1 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \\ -1 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \\ 0 \\ -1 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ -1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 0 \\ -1 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 1 \\ -1 \end{bmatrix} \right\}.$$

The bijection  $g : \mathcal{B}_2 \rightarrow \mathcal{B}_3$ , given by

$$g(x) = \begin{bmatrix} 1 & 1 & 1 \\ -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix} x,$$

preserves equalities in  $M_{\mathcal{B}}(y)$ , i.e.,  $y_{\alpha_1-\alpha_2} = y_{\alpha_3-\alpha_4}$  implies that  $y_{g(\alpha_1)-g(\alpha_2)} = y_{g(\alpha_3)-g(\alpha_4)}$ . Hence,  $\mathbf{L}(\mathcal{B}_3) = \mathbf{L}(\mathcal{B}_2)$ . In the sequel, we will use  $\mathbf{L}(\mathcal{B}_2)$  in favor of  $\mathbf{L}(\mathcal{B}_3)$ , due to its more compact representation. The equivalence  $\mathbf{L}(\mathcal{B}_3) = \mathbf{L}(\mathcal{B}_2)$  can also be understood as a consequence of Remark 3.1.  $\triangle$

We show now that, despite the smaller size of  $\mathcal{F}(\mathcal{B}_1)$  compared to  $\mathcal{F}(\mathcal{B}_2)$ , see (3.43), their induced approximations of  $\text{CUT}_\infty^4$  are equally strong. To do so, we define the following partial order.

**Definition 3.23.** Let  $\mathcal{B} \subseteq \mathbb{Z}^p$ , and let  $\mathcal{B}' \subseteq \mathcal{B}$ , with  $k := |\mathcal{B}'|$ . We say that  $\mathcal{B}$  extends  $\mathcal{B}'$ , denoted  $\mathcal{B}' \models \mathcal{B}$ , if and only if, for all  $X' \in \mathcal{F}(\mathcal{B}')$ , there exists an  $X \in \mathcal{F}(\mathcal{B})$  satisfying  $X_{1:k,1:k} = X'$ . Here, it is implicitly assumed that bases  $\mathcal{B}'$  and  $\mathcal{B}$  are ordered, and that the first  $k$  elements of  $\mathcal{B}$  are the elements of  $\mathcal{B}'$ , in the same order.

For a given  $X' \in \mathcal{F}(\mathcal{B}')$ , the problem of deciding whether there exists a  $X \in \mathcal{F}(\mathcal{B})$  that satisfies  $X_{1:k,1:k} = X'$  can be formulated as follows. Let  $(y_\alpha)_{\alpha \in \mathcal{B}' - \mathcal{B}'}$  be the sequence that satisfies  $X' = M_{\mathcal{B}'}(y)$ , see (3.41). Define the partially specified matrix  $X$ , with rows and columns indexed by the elements of  $\mathcal{B}$ , and entries given by

$$X_{\alpha,\beta} = \begin{cases} y_{\alpha-\beta} & \text{if } \alpha - \beta \in \mathcal{B}' - \mathcal{B}' \\ \star_{\alpha-\beta} & \text{else.} \end{cases} \quad \alpha, \beta \in \mathcal{B}. \quad (3.49)$$

The PSD completion problem [125] is to find values for  $\{\star_{\alpha-\beta}\}_{\alpha,\beta \in \mathcal{B}}$  that make  $X$  Hermitian PSD. If such values exist, the resulting fully specified  $X$  satisfies  $X \in \mathcal{F}(\mathcal{B})$  and  $X_{1:k,1:k} = X'$ , and we say that  $X$  is an extension of  $X'$ .

**Remark 3.24.** There is a small difference between the PSD completion problem presented here and the PSD completion problem from [125]. In our definition, it can occur that some unspecified entries of  $X$  are restricted to be equal. For example, in our definition, consider the case that  $\alpha, \beta, \gamma, \delta \in \mathcal{B}$  are such that  $\alpha - \beta = \gamma - \delta \notin \mathcal{B}' - \mathcal{B}'$ . Then both  $X_{\alpha,\beta}$  and  $X_{\gamma,\delta}$  are unspecified, and we are interested only in PSD completions that assign these two entries the same value  $X_{\alpha,\beta} = X_{\gamma,\delta} = \star_{\alpha-\beta}$ . In [125], none of the unspecified entries are restricted to be equal to one another. Whenever we use results from [125] in the sequel, this difference does not occur. In particular, this difference does not occur when  $|\mathcal{B}'| = |\mathcal{B}| - 1$ .  $\triangle$

It is not difficult to show the following implication

$$\mathcal{B}_1 \models \mathcal{B}_2 \implies \mathbf{L}(\mathcal{B}_1) = \mathbf{L}(\mathcal{B}_2), \quad (3.50)$$

see (3.44). We will use (3.50), in combination with PSD completion theory, to prove that the sets  $\mathbf{L}(\mathcal{B}_i)$  define equivalent relaxations of  $\mathcal{E}_\infty^4$ .

**Lemma 3.25.** For  $\mathcal{B}_1$  and  $\mathcal{B}_2$  as in (3.45) and  $\mathbf{L}(\mathcal{B}_i)$  as in (3.44), we have that  $\mathbf{L}(\mathcal{B}_1) = \mathbf{L}(\mathcal{B}_2)$ .

*Proof.* By (3.50), it suffices to show that  $\mathcal{B}_1 \models \mathcal{B}_2$ . Thus, we need to verify that for

all  $X' \in \mathcal{F}(\mathcal{B}_1)$ , the corresponding partially specified matrix

$$X = \begin{bmatrix} & & & & & & X'_{3,4} \\ & & & & & & X'_{3,6} \\ & & & X' & & & \star \\ & & & & & & X'_{3,1} \\ & & & & & & \star \\ & & & & & & X'_{3,2} \\ X'_{4,3} & X'_{6,3} & \star & X'_{1,3} & \star & X'_{2,3} & 1 \end{bmatrix} \in \mathcal{H}^7, \quad (3.51)$$

can be completed to a Hermitian PSD matrix, by finding (possibly distinct) values for  $\star$ . To derive the pattern of equalities in (3.51), we have used (3.46).

Note that the only unspecified entries of  $X$  are at position  $(3, 7)$  and  $(5, 7)$  (ignoring the lower triangular part of  $X$ ). We associate to this pattern of unspecified entries a graph  $\mathcal{G}$  of order 7, defined as

$$\begin{aligned} \mathcal{G} &= (V, E), \quad V = [7] \text{ and} \\ E &= \{\{i, j\} : i, j \in V, X_{ij} \neq \star\} = \{\{i, j\} : 1 \leq i < j \leq 7\} \setminus (\{3, 7\} \cup \{5, 7\}) \end{aligned} \quad (3.52)$$

Observe that  $\mathcal{G}$  is chordal. Then, by [125, Thm. 7],  $X$  can be completed to a PSD matrix if every fully specified principal submatrix of  $X$  (i.e., a principal submatrix that does not contain any  $\star$  values) is PSD. To investigate this condition, we write  $X_J$ ,  $J \subseteq [7]$ , for the principal submatrix of  $X$ , indexed by rows and columns in  $J$ . Before we consider all such fully specified submatrices  $X_J$ , we consider first  $Z_{\mathcal{J}}$ , for  $\mathcal{J} := \{1, 2, 4, 6, 7\}$ . Note that  $X_{\mathcal{J}}$  is fully specified, and given by

$$(X_{\mathcal{J}})_{ij} = L_{\mathbf{y}}(Z_{ij}), \text{ for } Z = \begin{bmatrix} 1 & \bar{x}_1 & \bar{x}_3 & x_1\bar{x}_3 & x_2\bar{x}_3 \\ x_1 & 1 & x_1\bar{x}_3 & x_1^2\bar{x}_3 & x_1x_2\bar{x}_3 \\ x_3 & \bar{x}_1x_3 & 1 & x_1 & x_2 \\ \bar{x}_1x_3 & \bar{x}_1^2x_3 & \bar{x}_1 & 1 & \bar{x}_1x_2 \\ \bar{x}_2x_3 & \bar{x}_1\bar{x}_2x_3 & \bar{x}_2 & x_1\bar{x}_2 & 1 \end{bmatrix},$$

and  $L_{\mathbf{y}}$  as in (3.47). Note that  $P^\top X_{\mathcal{J}} P = \bar{X}_{J'}$  for the permutation matrix  $P = E_{14} + E_{25} + E_{31} + E_{42} + E_{53}$  and  $J' = \{1, 2, 3, 4, 6\}$ . Thus, matrix  $X_{\mathcal{J}}$  is similar to  $\bar{X}_{J'}$ . Now, since  $X_{J'}$  is a fully specified submatrix of  $X'$ , and since  $X' \succeq 0$ , it follows that  $X_{J'} \succeq 0$ . This implies that  $\bar{X}_{J'} \succeq 0$ , which in turn implies that  $X_{\mathcal{J}} \succeq 0$ .

Let us now show that for any  $J \subseteq [7]$  such that  $X_J$  is fully specified,  $X_J \succeq 0$ . We distinguish two cases:

1.  $J \subseteq \mathcal{J}$ . Then  $X_J$  is a submatrix of  $X_{\mathcal{J}}$ , and therefore  $X_J \succeq 0$  since  $X_{\mathcal{J}} \succeq 0$ .
2.  $J \not\subseteq \mathcal{J}$ . Since  $J \not\subseteq \mathcal{J}$ ,  $3 \in J$  or  $5 \in J$ . As both  $X_{3,7}$  and  $X_{5,7}$  are unspecified, and  $X_J$  is fully specified, it follows that  $7 \notin J$ . Thus  $J \subseteq [6]$ . Consequently,  $X_J$  is a submatrix of  $X'$ , and  $X' \in \mathcal{F}(\mathcal{B}_1)$  implies  $X' \succeq 0$ , which shows that  $X_J \succeq 0$ .

To conclude, every fully specified principal submatrix of  $X$  is PSD, and the associated graph  $\mathcal{G}$  is chordal. By [125, Thm. 7], any  $X' \in \mathcal{F}(\mathcal{B}_1)$  can be extended to a fully specified matrix in  $X \in \mathcal{F}(\mathcal{B}_2)$  as in (3.51), which implies that  $\mathcal{B}_1 \models \mathcal{B}_2$ . By (3.50), this completes the proof.  $\square$

We now relate  $\mathbf{L}(\mathcal{B}_1)$ , see (3.44), to the second semidefinite lifting proposed in [154]. This lifting is given by  $\mathbf{L}(\mathcal{B}_4)$ , for

$$\mathcal{B}_4 := \left\{ \alpha \in \{0, 1, 2\}^3 : \sum_{i \in [3]} \alpha_i \leq 2 \right\}. \quad (3.53)$$

Note that  $|\mathcal{B}_4| = 10 > |\mathcal{B}_1| = 6$ . Despite this difference, the induced relaxations of  $\text{CUT}_\infty^4$  are equivalent, as shown in the following result.

**Theorem 3.26.** *For  $\mathcal{B}_4$  as in (3.53), we have that  $\mathbf{L}(\mathcal{B}_4) = \mathbf{L}(\mathcal{B}_1)$ , see (3.44).*

*Proof.* We start by considering the proof of Lemma 3.25 more abstractly. Let  $\mathcal{B}' \subseteq \mathcal{B} \subseteq \mathbb{Z}^p$ , where  $k := |\mathcal{B}'| = |\mathcal{B}| - 1$  (Note that this is the case for  $\mathcal{B}_1$  and  $\mathcal{B}_2$ , see Lemma 3.25). Consider the problem of extending some  $X' \in \mathcal{F}(\mathcal{B}')$  to some  $X \in \mathcal{F}(\mathcal{B})$ . This  $X$  should be thought of as in (3.49), possibly containing unspecified  $\star$  values. Let

$$\mathcal{J} := \{i \in \mathbb{N} : X_{i,k+1} \neq \star\},$$

so that matrix  $X_{\mathcal{J}}$ , the submatrix of  $X$  indicated by the elements of  $\mathcal{J}$ , is fully specified by  $X'$ . Note that the associated graph  $\mathcal{G}$ , see (3.52), is chordal and we may apply again [125, Thm. 7]. By similar reasoning as in Lemma 3.25, the condition that  $X_{\mathcal{J}}$  is similar to a submatrix of  $X'$ , is sufficient (although not necessary) for  $\mathcal{B}' \models \mathcal{B}$  to hold.

Following these steps for specific sets  $\mathcal{B}'$  and  $\mathcal{B}$ , we are able to prove the following relations, where we write  $\mathcal{B} \models \dots \cup \beta_1 \models \dots \cup \beta_2$  as shorthand for  $\mathcal{B} \models \mathcal{B} \cup \{\beta_1\} \models \mathcal{B} \cup \{\beta_1, \beta_2\}$ . Starting from

$$\mathcal{B}_5 = \left\{ \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} \right\},$$

we have (details omitted)

$$\mathcal{B}_5 \models \dots \cup \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} \models \dots \cup \begin{bmatrix} 2 \\ 0 \\ 0 \end{bmatrix} \models \dots \cup \begin{bmatrix} 0 \\ 2 \\ 0 \end{bmatrix} \models \dots \cup \begin{bmatrix} 0 \\ 0 \\ 2 \end{bmatrix} = \mathcal{B}_4, \quad (3.54)$$

$$\mathcal{B}_5 \models \dots \cup \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix} \models \dots \cup \begin{bmatrix} -1 \\ 1 \\ 0 \end{bmatrix} := \mathcal{B}_6, \quad (3.55)$$

and starting from  $\mathcal{B}_1$  as in (3.45), we have

$$\mathcal{B}_1 \models \cdots \cup \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} \models \cdots \cup \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} = \mathcal{B}_6. \quad (3.56)$$

Combining the implication (3.50) (which holds more generally for  $\mathcal{B}_i$  and  $\mathcal{B}_j$ ), with equations (3.55) and (3.56) yields that  $\mathbf{L}(\mathcal{B}_1) = \mathbf{L}(\mathcal{B}_5)$ . Since  $\mathbf{L}(\mathcal{B}_5) = \mathbf{L}(\mathcal{B}_4)$  by (3.54), the result follows.  $\square$

By combining results of Lemma 3.25 and Theorem 3.26, we obtain the following.

**Corollary 3.27.** *For all  $i, j \in [6]$ ,  $\mathbf{L}(\mathcal{B}_i) = \mathbf{L}(\mathcal{B}_j)$ .*

In the sequel, we will only refer to  $\mathbf{L}(\mathcal{B}_1)$  for compactness, as  $|\mathcal{B}_1| = \min_{i \in [6]} |\mathcal{B}_i|$ . Next, we show that  $\mathbf{L}(\mathcal{B}_1)$  does not contain any of the rank 2 extreme points of  $\mathcal{E}_\infty^4$ . Let us first characterize the set of rank 2 extreme points of  $\mathcal{E}_\infty^4$ . For this, we require the following definition, see matrix  $F$  from [194, Section 2.2].

**Definition 3.28.** We say that

$$G = \begin{bmatrix} x_1 & u_1 & w_1 & v_1 \\ x_2 & u_2 & w_2 & v_2 \end{bmatrix} \in \mathbb{C}^{2 \times 4}$$

is an Extremal Gram Factor (EGF) if and only if its columns  $x, u, w, v$  have norm 1, and the matrix

$$F := \begin{bmatrix} |x_1|^2 & x_1 \overline{x_2} & \overline{x_1} x_2 & |x_2|^2 \\ |u_1|^2 & u_1 \overline{u_2} & \overline{u_1} u_2 & |u_2|^2 \\ |w_1|^2 & w_1 \overline{w_2} & \overline{w_1} w_2 & |w_2|^2 \\ |v_1|^2 & v_1 \overline{v_2} & \overline{v_1} v_2 & |v_2|^2 \end{bmatrix} \quad (3.57)$$

is non-singular. Equivalently,  $\det(F) \neq 0$ .

Now  $P$ , the set of rank 2 extreme points of  $\mathcal{E}_\infty^4$ , is given by the product of EGFs. That is,

$$P := \{\text{rank 2 extreme points of } \mathcal{E}_\infty^4\} = \{G^H G : G \text{ is an EGF}\}, \quad (3.58)$$

as proven in [194] (note that EGFs are defined for general matrix sizes in [194]). Thus, if  $A$  is a rank 2 extreme point of  $\mathcal{E}_\infty^4$ , it must be of the form  $A = G^H G$ , where  $G$  is an EGF. Given such  $A$ , the corresponding matrix  $G$  is unique up to unitary transformation of its columns. We will use MATLAB like notation for indexing submatrices of  $G$ , i.e., for some  $J \subseteq [4]$ ,  $G_{:,J} \in \mathbb{C}^{2 \times |J|}$  denotes the submatrix obtained by taking all rows of  $G$ , and columns of  $G$  indexed by  $J$ .

Let us prove several results related to EGFs.

**Lemma 3.29.** *Let  $G \in \mathbb{C}^{2 \times 4}$  be an EGF. Then for any  $J \subseteq [4]$ ,  $|J| = 2$ , the  $2 \times 2$  submatrix  $G_{:,J}$  is invertible.*

*Proof.* Proof by contradiction: assume that  $G$  is an EGF, and that for some  $J \subseteq [n]$ ,  $|J| = 2$ , matrix  $G_{:,J} \in \mathbb{C}^{2 \times 2}$  is singular. A  $2 \times 2$  matrix is singular if and only if its second column equals its first column multiplied by some  $r \in \mathbb{C}$ . Since the columns of  $G$  have norm 1, we find that  $|r| = 1$ . But this implies that matrix  $F$ , see (3.57), has two identical rows, and is thus singular. This contradicts the assumption that  $G$  is an EGF.  $\square$

**Lemma 3.30.** *For any matrix  $A \in P$ , see (3.58), there exists an EGF  $G \in \mathbb{C}^{2 \times 4}$ , see Definition 3.28, that satisfies*

$$A = G^H G \text{ and } G = \begin{bmatrix} 1 & u_1 & w_1 & v_1 \\ 0 & u_2 & w_2 & v_2 \end{bmatrix} = [\mathbf{e} \quad u \quad w \quad v], \quad (3.59)$$

where  $u_2, w_2$ , and  $v_2$  are nonzero and  $\mathbf{e} = [1, 0]^T$ .

*Proof.* Since  $A \in P$ , there exists an EGF  $\tilde{G}$  such that  $A = \tilde{G}^H \tilde{G}$ . Let  $z := \tilde{G}_{:,1} \in \mathbb{C}^2$ , and consider the matrix  $Q := \begin{bmatrix} \bar{z}_1 & \bar{z}_2 \\ -z_2 & z_1 \end{bmatrix}$ . It is easy to see that  $Q$  is unitary, and  $Qz = \mathbf{e}$ . Then  $G := Q\tilde{G}$  is an EGF satisfying the properties of the lemma. Note that the entries  $u_2, w_2$  and  $v_2$  are nonzero, because each  $2$  by  $2$  submatrix of  $G$  is invertible (Lemma 3.29).  $\square$

In the sequel, we will thus only consider EGFs of the form (3.59). Note that this simplifies matrix  $F$  from (3.57). We are now ready to prove the following.

**Theorem 3.31.** *For  $P$  as in (3.58), we have that  $\mathbf{L}(\mathcal{B}_1) \cap P = \emptyset$ .*

*Proof.* Let  $A \in P$ . Then, without loss of generality,  $A = G^H G$ , where  $G$  is an EGF of the form (3.59). Proof by contradiction: suppose that  $A \in \mathbf{L}(\mathcal{B}_1)$ . Then  $\exists Z \in \mathcal{F}(\mathcal{B}_1)$ , see (3.43), satisfying  $Z_{1:4,1:4} = A$ . Let  $\ell \in \{5, 6\}$  and denote by  $Z_\ell$  the  $5 \times 5$  principal submatrix of  $Z$ , with rows and columns indexed by  $[4] \cup \ell$ . Since  $Z_\ell \succeq 0$ , there exists a matrix  $G_\ell$  such that  $Z_\ell = G_\ell^H G_\ell$ . We may assume that  $G_\ell$  is of the form

$$G_\ell = \begin{bmatrix} G & z_\ell \\ \mathbf{0}_4^T & \alpha_\ell \end{bmatrix} \in \mathbb{C}^{3 \times 5}, \text{ with } z_\ell \in \mathbb{C}^2, \alpha_\ell \in \mathbb{C} \text{ and } z_\ell^H z_\ell + |\alpha_\ell|^2 = 1. \quad (3.60)$$

Note that the last column of  $Z_\ell$  is then given by  $[z_\ell^H G \quad 1]^H$ . Moreover, for each  $\ell \in \{5, 6\}$ , precisely two of the entries in  $G^H z_\ell$  are determined by  $A$ . For example, if  $\ell = 5$ , then we have

$$G_{:,\{1,3\}}^H z_5 = \begin{bmatrix} u^H w \\ u^H \mathbf{e} \end{bmatrix}, \text{ with } G_{:,\{1,3\}}^H = [\mathbf{e} \quad w]^H = \begin{bmatrix} 1 & 0 \\ \bar{w}_1 & \bar{w}_2 \end{bmatrix}. \quad (3.61)$$

The equations (3.61) follow from the pattern of equalities in (3.46). In particular,  $(G^H z_5)_1 = L_y(x_1 \bar{x}_2) = A_{2,3} = (G^H G)_{2,3} = u^H w$ .

By Lemma 3.29,  $G_{:,\{1,3\}}^H$  is invertible, and hence  $z_5$  is uniquely determined by (3.61). Specifically,

$$z_5 = \left( G_{:,\{1,3\}}^H \right)^{-1} \begin{bmatrix} u^H w \\ u^H \mathbf{e} \end{bmatrix} = \begin{bmatrix} u^H w \\ (\bar{w}_1 - \bar{w}_1 u^H w) / \bar{w}_2 \end{bmatrix}. \quad (3.62)$$

We now claim that  $\|z_5\| = 1$ , in which case  $\alpha_5 = 0$ , by (3.60). To verify this claim, we compute first

$$\operatorname{Re}(u_1 \bar{w}_1 u^H w) = |u_1|^2 |w_1|^2 + \operatorname{Re}(u_1 \bar{w}_1 \bar{u}_2 w_2), \quad (3.63)$$

which is a term appearing in the computation of  $\|z_5\|^2 = z_5^H z_5$ . Then, by combining (3.62) and (3.63), we find

$$\begin{aligned} z_5^H z_5 &= |u^H w|^2 + \frac{|u_1|^2 + |w_1|^2 |u^H w|^2 - 2 \operatorname{Re}(u_1 \bar{w}_1 u^H w)}{|w_2|^2} \\ &= \frac{|u_1|^2 + |u^H w|^2 - 2 \operatorname{Re}(u_1 \bar{w}_1 u^H w)}{|w_2|^2} \\ &= \frac{|u_1|^2 + |u_1|^2 |w_1|^2 + |u_2|^2 |w_2|^2 + 2 \operatorname{Re}(\bar{u}_1 w_1 u_2 \bar{w}_2) - 2 \operatorname{Re}(u_1 \bar{w}_1 u^H w)}{|w_2|^2} \\ &= \frac{|u_1|^2 - |u_1|^2 |w_1|^2 + (1 - |u_1|^2)(1 - |w_1|^2)}{|w_2|^2} = 1, \end{aligned}$$

where we have also used that  $|u_1|^2 + |u_2|^2 = |w_1|^2 + |w_2|^2 = 1$ .

Vector  $z_6$  satisfies the system

$$G_{:, \{1,4\}}^H z_6 = \begin{bmatrix} u^H v \\ u^H \mathbf{e} \end{bmatrix}, \quad \text{with } G_{:, \{1,4\}}^H = [\mathbf{e} \quad v]^H = \begin{bmatrix} 1 & 0 \\ \bar{v}_1 & \bar{v}_2 \end{bmatrix}.$$

which is similar to (3.61). It is therefore also straightforward to show that the vector

$$z_6 = \left( G_{:, \{1,4\}}^H \right)^{-1} \begin{bmatrix} u^H v \\ u^H \mathbf{e} \end{bmatrix} = \begin{bmatrix} u^H v \\ (\bar{u}_1 - \bar{v}_1 u^H v) / \bar{v}_2 \end{bmatrix} \quad (3.64)$$

satisfies  $\|z_6\| = 1$ . This implies that  $Z$  is of the form

$$Z = V^H V, \quad \text{for } V := [\mathbf{e} \quad u \quad w \quad v \quad z_5 \quad z_6] \in \mathbb{C}^{2 \times 6}. \quad (3.65)$$

Now  $Z \in \mathcal{F}(\mathcal{B}_1) \implies Z_{5,6} = Z_{3,4} = w^H v$ , see (3.46), and (3.65) implies that  $Z_{5,6} = z_5^H z_6$ . Therefore,

$$w^H v = z_5^H z_6 = w^H v + \frac{\det(F)}{w_2 \bar{v}_2}, \quad (3.66)$$

for  $F$  as in (3.57). The second equality in (3.66) follows from substituting (3.62) and (3.64), and has been verified by a computer algebra system.

Equation (3.66) implies that  $\det(F) = 0$ , which contradicts the fact that  $G$  is an EGF (Definition 3.28).  $\square$

We now show that  $\text{CUT}_\infty^n$  contains all rank 2 points of  $\mathcal{E}_\infty^n$ , if these are not extreme. To prove this result, we require an intermediate result and the notion of a *perturbation*, see [194]. We say that a nonzero Hermitian matrix  $B$  is a perturbation of some  $A \in \mathcal{E}_\infty^n$ , if there exists some  $t > 0$  such that  $A \pm tB \in \mathcal{E}_\infty^n$ . Thus, if  $A$  admits some perturbation  $B$ , it is not an extreme point of  $\mathcal{E}_\infty^n$ . Additionally, if  $A = G^H G$ , then any perturbation is of the form  $B = G^H R G$  [194, Thm. 1(a)], with  $\operatorname{diag}(B) = \mathbf{0}$  and  $R$  Hermitian.

**Remark 3.32.** The notion of a perturbation can be generalized to sets of the form

$$\{X \in \mathcal{H}_+^n : \langle A_i, X \rangle = b_i, i \in [m], \langle A'_i, X \rangle \leq b'_i, i \in [m']\},$$

for  $A_1, \dots, A_m, A'_1, \dots, A'_{m'} \in \mathcal{H}^n$ ,  $b \in \mathbb{R}^m$  and  $b' \in \mathbb{R}^{m'}$ . See e.g., [74, Thm. 31.5.3] or [189, Thm. 2.1].  $\triangle$

The following result, which is proven using perturbations, is a generalization of the sufficiency part of [194, Cor. 4]. We suspect that this result is known, although we were unable to find a reference.

**Lemma 3.33.** *Let  $A \in \mathcal{E}_\infty^n$ , with  $\text{rk}(A) = r$ . Then, there exist extreme points  $A_1, \dots, A_r$  of  $\mathcal{E}_\infty^n$ , such that  $A \in \text{Conv}\{A_1, \dots, A_r\}$ , and  $\text{rk}(A_i) \leq \text{rk}(A)$  for all  $i \in [r]$ . If  $A$  is not an extreme point of  $\mathcal{E}_\infty^n$ , then we have that  $\text{rk}(A_i) < \text{rk}(A)$  for all  $i \in [r]$ .*

*Proof.* See Page 206 in Appendix B.2.  $\square$

Lemma 3.33 leads directly to the following similar result regarding  $\text{CUT}_\infty^n$ .

**Theorem 3.34.** *Let  $A \in \mathcal{E}_\infty^n$  and  $r := \text{rk}(A)$ . If  $n \geq 2$ ,  $r = 2$ , and  $A$  is not an extreme point of  $\mathcal{E}_\infty^n$ , then  $A$  is the convex combination of two extreme points of  $\text{CUT}_\infty^n$ . If  $n \in \{2, 3\}$  and  $1 \leq r \leq n$ , then  $A$  is the convex combination of  $r$  extreme points of  $\text{CUT}_\infty^n$ .*

*Proof.* For the first case, it follows from Lemma 3.33 that  $A \in \text{Conv}\{A_1, A_2\}$ , where the matrices  $A_i$ ,  $i \in [2]$ , are extreme points of  $\mathcal{E}_\infty^n$ , with  $\text{rk}(A_i) < \text{rk}(A)$ . Since  $\text{rk}(A) = 2$ , we have that  $\text{rk}(A_i) = 1$ , which implies that the matrices  $A_i$  are also extreme points of  $\text{CUT}_\infty^n$ . The second case also follows from Lemma 3.33, and the fact that  $\mathcal{E}_\infty^n = \text{CUT}_\infty^n$  for  $n \in \{2, 3\}$ , see Lemma 3.21.  $\square$

It is known that any  $A \in \text{CUT}_\infty^n$  can be written as a convex combination of at most  $n^2 - n + 1$  extreme points of  $\text{CUT}_\infty^n$  [150, Lem. 3], which follows from Carathéodory's theorem. It is stated in [150] that 'a smaller bound would help in reducing the size of the problem for finding a nearest matrix in  $\text{CUT}_\infty^n$ '. Theorem 3.34 offers some help in reducing this bound, but only for restricted values of  $n$  and  $r$ . When  $n = 4$ , we can slightly strengthen Theorem 3.34.

**Lemma 3.35.** *Let  $A$  be a non-extreme point of  $\mathcal{E}_\infty^4$  of rank  $r > 1$ . There exist  $k$ ,  $k \leq r$ , extreme points  $A_1, \dots, A_k$  of  $\mathcal{E}_\infty^4$ , such that  $A \in \text{Conv}\{A_1, \dots, A_k\}$ ,  $\sum_{i=1}^k \text{rk}(A_i) = r$  and at least one of the  $A_i$ ,  $i \in [k]$ , satisfies  $\text{rk}(A_i) = 1$ .*

*Proof.* See Page 206 in Appendix B.2.  $\square$

Let us now return to the case  $n = 4$ , specifically the relation between  $\text{CUT}_\infty^4$  and  $\mathbf{L}(\mathcal{B}_1)$  (the set  $\mathbf{L}(\mathcal{B}_1)$  is defined in (3.44) on Page 75). We have that

$$\{X \in \mathbf{L}(\mathcal{B}_1) : \text{rk}(X) \in \{1, 2\}\} \subseteq \text{CUT}_\infty^4 \subseteq \mathbf{L}(\mathcal{B}_1). \quad (3.67)$$

The first inclusion in (3.67) follows from Theorems 3.31 and 3.34. The second inclusion follows from (3.48). Numerical tests, see also [150], and (3.67) lead us to the following conjecture:

**Conjecture 3.36.** The second semidefinite lifting is exact for  $\text{CUT}_\infty^4$ , i.e.,  $\mathbf{L}(\mathcal{B}_1) = \text{CUT}_\infty^4$ .

**Remark 3.37.** Conjecture 3.36 can be connected to the notion of a flat extension from the literature, see e.g., [64]. The term extension is related to the definition of extension as given in Definition 3.23. In Definition 3.23, the larger matrix  $X$  is considered an extension of the smaller matrix  $X'$ . Matrix  $X$  is said to be a flat extension of  $X'$  if, along with the properties outlined in Definition 3.23, we also have  $\text{rk}(X) = \text{rk}(X')$ . In [154], there are several results related to such flat extensions, specifically for the bases  $\mathbb{N}_r^p := \{\alpha \in \mathbb{N}^p : \sum_{i=1}^p \alpha_i \leq r\}$ . Now [154, Thm. 5.2] leads to an equivalent reformulation of Conjecture 3.36. Namely, Conjecture 3.36 is true if and only if all  $X' \in \mathbf{L}(\mathcal{B}_1) \subseteq \mathcal{E}_\infty^n$  admit a flat extension  $X \in \mathcal{F}(\mathbb{N}_3^3)$ , for  $\mathcal{F}(\cdot)$  as in (3.43).  $\triangle$

We show now that all  $X \in \mathbf{L}(\mathcal{B}_1)$  satisfy a valid inequality for  $\text{CUT}_\infty^4$ , found by the authors of [150]. This inequality is given as follows:

$$\langle H, X \rangle \leq 6 \quad \forall X \in \text{CUT}_\infty^4, \text{ where } H := \begin{bmatrix} 0 & -\mathbf{i} & \mathbf{i} & 1 \\ \mathbf{i} & 0 & -\mathbf{i} & 1 \\ -\mathbf{i} & \mathbf{i} & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}. \quad (3.68)$$

In [150, Section 4.4], it is also shown that  $\max_{X \in \mathcal{E}_\infty^4} \langle H, X \rangle \geq 6.9282\dots$ . Both this inequality and inequality (3.68) are proven numerically in [150]. We provide analytical proofs of both inequalities in the following lemma. We also extend the results from  $m = \infty$  to integer  $m$ , and we prove that  $\max_{X \in \mathbf{L}(\mathcal{B}_1)} \langle H, X \rangle = 6$ , for  $\mathbf{L}(\mathcal{B}_1)$  defined in (3.44). This proves that the matrices in  $\mathbf{L}(\mathcal{B}_1)$  also satisfy inequality (3.68).

**Lemma 3.38.** *Let  $H$  be as in (3.68). For all integers  $m \geq 3$  or  $m = \infty$ , we have that*

1.  $\max_{X \in \mathcal{E}_m^4} \langle H, X \rangle = 4\sqrt{3} \approx 6.928$ .
2.  $\max_{X \in \text{CUT}_m^4} \langle H, X \rangle = \max_{X \in \mathbf{L}(\mathcal{B}_1)} \langle H, X \rangle = 6$ .

*Proof.* See Page 207 in Appendix B.2.  $\square$

Additionally, the matrices in  $\mathbf{L}(\mathcal{B}_1)$  also satisfy all the infinite cuts that are ROC equivalent to the inequality  $\langle H, X \rangle \leq 6$ , see Lemma 3.6. Using Lemma 3.38, we can easily compute the strength, see (3.8), of this inequality.

**Corollary 3.39.** *Let  $H$  be as in (3.68). Then, for all integers  $m \geq 3$  or  $m = \infty$ , we have that  $\text{str}(H, m) = \frac{2}{\sqrt{3}} \approx 1.155$ .*

**Remark 3.40.** Inequality (3.68) is not a gap inequality, see Lemma 3.9. This can be shown by using methods similar to those used in Remark 3.15.  $\triangle$

To conclude this section, we provide a generalization of Theorem 3.26 for any  $n \geq 2$ . We define, for  $n \geq 2$ , the bases

$$\widetilde{\mathcal{A}}^n := \left\{ \alpha \in \{0, 1\}^{n-1} : \sum_{i=1}^{n-1} \alpha_i \leq 2 \right\} \subsetneq \mathcal{A}^n := \left\{ \alpha \in \{0, 1, 2\}^{n-1} : \sum_{i=1}^{n-1} \alpha_i \leq 2 \right\},$$

where the first  $n$  elements of both  $\widetilde{\mathcal{A}}^n$  and  $\mathcal{A}^n$  are  $\{\mathbf{0}_{n-1}\}$  and the  $n-1$  unit vectors, i.e., the columns of  $\mathbf{I}_{n-1}$ . Sets  $\mathcal{F}(\widetilde{\mathcal{A}}^n)$  and  $\mathcal{F}(\mathcal{A}^n)$  are defined analogously to (3.43). Note that  $\mathcal{A}^4 = \mathcal{B}_4$ , for  $\mathcal{B}_4$  as in (3.53). The bases  $\widetilde{\mathcal{A}}^n$  and  $\mathcal{A}^n$  can be used to approximate  $\text{CUT}_\infty^n$ . If we define the sets

$$\mathbf{L}^n(\widetilde{\mathcal{A}}) = \left\{ X \in \mathcal{E}_\infty^n : \exists Z \in \mathcal{F}(\widetilde{\mathcal{A}}^n) \text{ satisfying } Z_{1:n,1:n} = X \right\}, \quad n \geq 2, \quad (3.69)$$

and similarly  $\mathbf{L}^n(\mathcal{A})$ , then  $\text{CUT}_\infty^n \subseteq \mathbf{L}^n(\mathcal{A}) \subseteq \mathbf{L}^n(\widetilde{\mathcal{A}}) \subseteq \mathcal{E}_\infty^n$ . We are now ready to present the following result.

**Lemma 3.41.** *For  $n \geq 2$ , we have that  $\mathbf{L}^n(\widetilde{\mathcal{A}}) = \mathbf{L}^n(\mathcal{A})$ .*

*Proof.* See Page 209 in Appendix B.2. □

**Remark 3.42.** Lemma 3.41 allows us to choose a smaller monomial basis, namely a basis without squared variables, without weakening the corresponding CSDP relaxation. There are several results in the literature on choosing a (smaller) monomial basis, though they are limited to real variables. For unconstrained polynomial optimization, the Newton polytope (see e.g., [203, Section III.A]) offers a monomial basis which is guaranteed to find sum of squares decompositions of sum of squares polynomials. Another, more broadly applicable, method is proposed in [307, Algorithm 4.1]. In contrast to this chapter, it is not proven in [307] that the basis returned by that algorithm offers the same relaxation strength as the standard basis of monomials.  $\triangle$

## 3.6 Extreme points of $\mathcal{E}_m^3$

In this section we derive necessary and sufficient conditions for a matrix to be an extreme rank 2 point of  $\mathcal{E}_m^3$ ,  $m > 2$  finite. For any such  $m$ , we provide an explicit rank 2 extreme point of  $\mathcal{E}_m^3$  (Lemma 3.46). Further, we extend this result for any finite  $n$  and  $m$ , which proves the strict inclusion of  $\text{CUT}_m^n$  in  $\mathcal{E}_m^n$  (Corollary 3.47).

For  $m > 2$ , we consider a general rank 2 matrix, parameterized as

$$N = \begin{bmatrix} 1 & N_{12} & N_{13} \\ \bar{N}_{12} & 1 & N_{23} \\ \bar{N}_{13} & \bar{N}_{23} & 1 \end{bmatrix} = G^H G \in \mathcal{E}_m^3, \text{ for } G = [\mathbf{e} \quad u \quad v] = \begin{bmatrix} 1 & u_1 & v_1 \\ 0 & u_2 & v_2 \end{bmatrix}, \quad (3.70)$$

where  $\|u\| = \|v\| = 1$ . We assume that at least one of  $u_2$  and  $v_2$  is nonzero to ensure that  $\text{rk}(N) = 2$ . Note that the parametrization (3.70) always exists, see e.g., Lemma 3.30 and [194]. We investigate under what conditions  $N$  is an extreme point of  $\mathcal{E}_m^3$ .

A perturbation of  $N \in \mathcal{E}_m^3$  (with respect to  $\mathcal{E}_m^3$ ) is a nonzero Hermitian matrix  $B = G^H R G$ , satisfying  $\text{diag}(B) = \mathbf{0}_3$ ,  $R \in \mathcal{H}^2$ , and for which there exists a  $t > 0$  such that  $N \pm tB \in \mathcal{E}_m^3$ , see [194] and also Section 3.5. The constraint  $\text{diag}(B) = \mathbf{0}_3$  implies  $\mathbf{e}^H R \mathbf{e} = R_{11} = 0$ , and  $u^H R u = v^H R v = 0$ . The system of these 3 equalities may be written in the following form:

$$R = \begin{bmatrix} 0 & \bar{\alpha} \\ \alpha & c \end{bmatrix}, \quad \begin{bmatrix} u_1 \bar{u}_2 & \bar{u}_1 u_2 & |u_2|^2 \\ v_1 \bar{v}_2 & \bar{v}_1 v_2 & |v_2|^2 \end{bmatrix} \begin{bmatrix} \alpha \\ \bar{\alpha} \\ c \end{bmatrix} = 0 \text{ for } \alpha \in \mathbb{C}, c \in \mathbb{R}. \quad (3.71)$$

Note the similarity with (3.57). Any possible perturbation  $B$  of  $N$  is of the following form

$$B = \begin{bmatrix} 0 & b_{12} & b_{13} \\ \bar{b}_{12} & 0 & b_{23} \\ \bar{b}_{13} & \bar{b}_{23} & 0 \end{bmatrix} = G^H R G = G^H \begin{bmatrix} 0 & \bar{\alpha} \\ \alpha & c \end{bmatrix} G, \quad \alpha \in \mathbb{C}, c \in \mathbb{R}. \quad (3.72)$$

Recall that  $N$  is not an extreme point of  $\mathcal{E}_m^3$  if and only if it admits a perturbation. There exist however simple sufficient conditions that show that a matrix  $N$  is not an extreme point. These conditions involve the interior of  $\text{Conv}(\mathcal{U}_m)$  and the boundary of  $\text{Conv}(\mathcal{U}_m)$ . We denote the boundary by

$$\partial\text{Conv}(\mathcal{U}_m) := \left\{ tu + (1-t)e^{2\pi i/m}u : t \in [0, 1], u \in \mathcal{U}_m \right\}.$$

We also require the related set

$$\partial\text{Conv}(\mathcal{U}_m) \setminus \mathcal{U}_m = \{u \in \partial\text{Conv}(\mathcal{U}_m) : u \notin \mathcal{U}_m\}. \quad (3.73)$$

**Lemma 3.43.** *Let  $m > 2$  be finite, and  $N \in \mathcal{E}_m^3$  with  $\text{rk}(N) = 2$ . If all the off-diagonal elements of  $N$  are interior points of  $\text{Conv}(\mathcal{U}_m)$ , or any off-diagonal element of  $N$  is contained in  $\mathcal{U}_m$ , then  $N$  is not an extreme point of  $\mathcal{E}_m^3$ .*

*Proof.* By Lemma 3.21,  $N$  is not an extreme point of  $\mathcal{E}_\infty^3$ . Thus,  $N$  admits some perturbation matrix  $B$  with respect to  $\mathcal{E}_\infty^3$ , i.e., there exists some  $t^* > 0$  such that  $N \pm tB \in \mathcal{E}_\infty^3$  for all  $t \in [0, t^*]$ . If all off-diagonal elements of  $N$  are interior points of  $\text{Conv}(\mathcal{U}_m)$ , then there exists some  $t \in [0, t^*]$  small enough such that  $N \pm tB \in \mathcal{E}_m^3$ , and the result follows.

Let us now assume that  $N$  has at least one upper-triangular off-diagonal element contained in  $\mathcal{U}_m$ . Then, without loss of generality, we have

$$N = G^H G, \quad \text{for } G = \begin{bmatrix} 1 & \kappa & u_1 \\ 0 & 0 & u_2 \end{bmatrix},$$

where  $\kappa \in \mathcal{U}_m$ . The off-diagonal elements of  $N$  are given by  $\kappa$ ,  $u_1$ , and  $\bar{\kappa}u_1$  and their complex conjugates. We distinguish three cases, based on the complex number  $u_1$ :

1.  $u_1 \in \mathcal{U}_m$ . Since  $\|u\| = 1$ , we have that  $|u_2|^2 = 1 - |u_1|^2 = 0$ , so that  $u_2 = 0$ . Then  $\text{rk}(N) = 1$ , which is not possible since we assumed that  $\text{rk}(N) = 2$ .
2.  $u_1$  is an interior point of  $\text{Conv}(\mathcal{U}_m)$ . Again, there exists a perturbation matrix  $B$  and  $t^* > 0$  such that  $N \pm tB \in \mathcal{E}_\infty^3$  for all  $t \in [0, t^*]$ . Note that, since  $N_{12} = \kappa \in \mathcal{U}_\infty$ ,  $B_{12} = 0$ . Note that the other off-diagonal elements of  $N$  are all interior points of  $\text{Conv}(\mathcal{U}_m)$ . Thus, there exists some small enough  $t \in [0, t^*]$  such that  $N \pm tB \in \mathcal{E}_m^3$ , and hence,  $N$  is not an extreme point of  $\mathcal{E}_m^3$ .

3.  $u_1 \in \partial\text{Conv}(\mathcal{U}_m) \setminus \mathcal{U}_m$ , see (3.73). Then  $u_1$  can be written as  $u_1 = w\delta + (1-w)\eta$ , where  $w \in (0, 1)$  and  $\delta, \eta$  are distinct  $m$ -roots of unity.

$$\begin{aligned} N &= \begin{bmatrix} 1 & \kappa & w\delta + (1-w)\eta \\ \bar{\kappa} & 1 & w\bar{\kappa}\delta + (1-w)\bar{\kappa}\eta \\ w\bar{\delta} + (1-w)\bar{\eta} & w\kappa\bar{\delta} + (1-w)\kappa\bar{\eta} & 1 \end{bmatrix} \\ &= w \begin{bmatrix} 1 \\ \bar{\kappa} \\ \bar{\delta} \end{bmatrix} \begin{bmatrix} 1 \\ \bar{\kappa} \\ \bar{\delta} \end{bmatrix}^H + (1-w) \begin{bmatrix} 1 \\ \bar{\kappa} \\ \bar{\eta} \end{bmatrix} \begin{bmatrix} 1 \\ \bar{\kappa} \\ \bar{\eta} \end{bmatrix}^H, \end{aligned}$$

so that clearly,  $N$  is not an extreme point of  $\mathcal{E}_m^3$ . □

It follows from Lemma 3.43 that any rank 2 extreme point of  $\mathcal{E}_m^3$  must have at least one element which is contained in the set (3.73), and its off-diagonal elements cannot be contained in  $\mathcal{U}_m$ . This allows us to characterize rank 2 extreme points of  $\mathcal{E}_m^3$ . We first require the following preparatory lemma.

**Lemma 3.44.** *Let  $m > 2$  be finite and  $N \in \mathcal{E}_m^3$  be a rank 2 matrix with entries  $N_{ij}$ . Let*

$$K := \{\{i, j\} \in [3] \times [3] : N_{ij} \in \partial\text{Conv}(\mathcal{U}_m) \setminus \mathcal{U}_m\}$$

and  $f : K \rightarrow [m]$  be the function that satisfies  $\text{Re}(\bar{\nu}_{f(ij)}N_{ij}) = \cos(\pi/m)$ , for  $\nu$  as in (3.6). If  $K \neq \emptyset$ , then any possible perturbation  $B$  of  $N$  must satisfy  $\text{Re}(\bar{\nu}_{f(ij)}B_{ij}) = 0$  for all  $\{i, j\} \in K$ .

*Proof.* Suppose that  $B$  is a perturbation of  $N$  (with respect to  $\mathcal{E}_m^3$ ) and  $\{i, j\} \in K$ . Then by definition of a perturbation, we must have  $N \pm tB \in \mathcal{E}_m^3$  for some  $t > 0$ . In particular,  $(N \pm tB)_{ij} \in \text{Conv}(\mathcal{U}_m)$ . Considering (3.6), this implies that

$$\begin{aligned} \text{Re}(\bar{\nu}_{f(ij)}(N \pm tB)_{ij}) \leq \cos\left(\frac{\pi}{m}\right) &\implies \cos\left(\frac{\pi}{m}\right) \pm t \text{Re}(\bar{\nu}_{f(ij)}B_{ij}) \leq \cos\left(\frac{\pi}{m}\right) \\ \implies \pm t \text{Re}(\bar{\nu}_{f(ij)}B_{ij}) \leq 0 &\implies \text{Re}(\bar{\nu}_{f(ij)}B_{ij}) = 0. \quad \square \end{aligned}$$

We now present the characterization of rank 2 extreme points of  $\mathcal{E}_m^3$ .

**Proposition 3.45.** *Let  $m, N, K$  and  $f$  be as in Lemma 3.44. Matrix  $N$  is an extreme point of  $\mathcal{E}_m^3$ , if and only if the following two conditions hold:*

1.  $K \neq \emptyset$ ;
2. There does not exist a perturbation  $B$  of  $N$  satisfying  $\text{Re}(\bar{\nu}_{f(ij)}B_{ij}) = 0$  for all  $\{i, j\} \in K$ .

*Proof.* ( $\implies$ ) Let  $N$  be a rank 2 extreme point of  $\mathcal{E}_m^3$ . By Lemma 3.43, not all off-diagonal elements of  $N$  can be in the interior of  $\text{Conv}(\mathcal{U}_m)$ , and none of the off-diagonal elements can be contained in  $\mathcal{U}_m$ . Thus  $K \neq \emptyset$ , satisfying Item 1. Since  $N$  is an extreme point, it does not admit a perturbation. In particular, it does not admit a perturbation that satisfies  $\text{Re}(\bar{\nu}_{f(ij)}B_{ij}) = 0 \forall \{i, j\} \in K$ , so Item 2 is satisfied.

( $\Leftarrow$ ) Let  $N \in \mathcal{E}_m^3$  be a rank 2 matrix and  $K \neq \emptyset$ . Lemma 3.44 states that any possible perturbation of a rank 2 matrix must satisfy  $\operatorname{Re}(\bar{v}_{f(ij)} B_{ij}) = 0 \forall \{i, j\} \in K$  when  $K \neq \emptyset$ . Because  $N$  satisfies Item 2, such a perturbation cannot exist. Thus,  $N$  admits no perturbation, and hence, is an extreme point.  $\square$

Using Proposition 3.45, we determine a rank 2 extreme point of  $\mathcal{E}_m^3$ , for any  $2 < m < \infty$ .

**Lemma 3.46.** *Fix some integer  $2 < m < \infty$ , and set*

$$N := G^H G, \text{ for } G := \begin{bmatrix} 1 & \frac{1}{2} + \frac{1}{2} \exp(2\pi \mathbf{i}/m) & \sin(\pi/m) \\ 0 & \sin(\pi/m) & \frac{1}{2} + \frac{1}{2} \exp(2\pi \mathbf{i}/m) \end{bmatrix}. \quad (3.74)$$

*Then  $N$  is a rank 2 extreme point of  $\mathcal{E}_m^3$ .*

*Proof.* See Page 210 in Appendix B.2.  $\square$

Using Lemma 3.46, we can directly show the following.

**Corollary 3.47.** *For finite  $m$  and  $n$ ,  $m \geq 2$  and  $n \geq 3$ , we have  $\operatorname{CUT}_m^n \subsetneq \mathcal{E}_m^n$ .*

*Proof.* For the case  $m = 2$  and  $n = 3$ , consider the matrix  $N = \frac{3}{2} \mathbf{I}_3 - \frac{1}{2} \mathbf{J}_3$ . Since  $N$  is real and PSD,  $N \in \mathcal{E}_2^3$ . Since  $N$  does not satisfy the triangle inequality  $N_{12} + N_{13} + N_{23} \geq -1$ , see (3.13),  $N \notin \operatorname{CUT}_2^3$ .

Lemma 3.46 proves that  $\operatorname{CUT}_m^n \subsetneq \mathcal{E}_m^n$  for all finite  $m > 2$  and  $n = 3$ . The case  $m > 2$  and  $n > 3$  follows by considering

$$\tilde{N} = \begin{bmatrix} N & \mathbf{0}_{3 \times (n-3)} \\ \mathbf{0}_{(n-3) \times 3} & \mathbf{I}_{n-3} \end{bmatrix} \in \mathcal{E}_m^n \setminus \operatorname{CUT}_m^n, \quad (3.75)$$

for  $N$  as in (3.74). The same extension as (3.75) for  $N = \frac{3}{2} \mathbf{I}_3 - \frac{1}{2} \mathbf{J}_3$  shows that  $\operatorname{CUT}_2^n \subsetneq \mathcal{E}_2^n$  for  $n > 3$ .  $\square$

## 3.7 Numerical results

In this section, we provide some computational results related to the previous sections. All CSDPs are first reformulated to equivalent real SDPs using YALMIP [202]. These real SDPs are then solved using MOSEK [228] with default settings on a server with Intel Xeon Gold 6126 CPU, running at 2.60GHz, with 512 GB RAM and using 8 cores.

### 3.7.1 Strength of cuts

We provide the numerical values of  $\mathbf{str}$  for the valid inequalities stated in Propositions 3.10 and 3.12, and (3.68). To provide a fair comparison, we have ensured that each matrix  $Q$  satisfies  $\langle Q, \mathbf{I} \rangle = 0$ , see also Remark 3.4.

Results are provided in Table 3.2. Strength values that have not been analytically computed in the previous sections, have now been computed using MOSEK [228]. The strength of the cuts in Proposition 3.10 tend to 1 as  $m \rightarrow \infty$ . For  $m = 3$ , the strongest cut is given by the facet defining inequalities from Proposition 3.12.

str of the cut as in:

$m$	Prop. 3.10, $n=3$	Prop. 3.10, $n=4$	Prop. 3.12, $n=3$	Eq. (3.68), $n=4$
2	1.500	1	1	1
3	1	1.333	1.815	1.155
4	1.500	1	1.169	1.155
5	1.146	1.038	1.077	1.155
6	1	1	1.075	1.155
7	1.114	1.010	1.011	1.155
8	1.061	1	1	1.155
9	1	1.004	1	1.155
$\infty$	1	1	1	1.155

Table 3.2: Numerical values of the strength **str** of various cuts.

### 3.7.2 Random objective functions

We consider the following optimization problem

$$\max_{X \in K_m} \langle Q, X \rangle, \quad (3.76)$$

for  $K_m = \mathcal{E}_m^n$  or  $K_m = \mathbf{T}(\mathcal{E}_m^n)$ , and  $m \in \{3, 4\}$ . Here  $Q \in \mathcal{H}^n$ ,  $\text{Diag}(Q) = \mathbf{0}$ , and  $\text{Im}(Q) \neq \mathbf{0}$ . The complex elliptope  $\mathcal{E}_m^n$  is defined in (3.3), and  $\mathbf{T}(\mathcal{E}_3^n)$  in (3.36). The set  $\mathbf{T}(\mathcal{E}_4^n)$  is defined as the set of matrices in  $\mathcal{E}_4^n$  for which each  $3 \times 3$  submatrix satisfies (3.25), the 16 (ROC equivalent) facet defining inequalities from Proposition 3.10, see Remark 3.13.

We set  $n = 100$ , and generate 250 matrices  $Q$  per value of  $m$  in the following way. Upper triangular entries of a matrix  $Q$  are of the form  $a + bi$ , where  $a$  and  $b$  are independent random integer variables, drawn uniformly from the set  $\{-10, -9, \dots, 9, 10\}$ . For each such  $Q$ , we solve (3.76) for  $K_m = \mathcal{E}_m^{100}$  and  $K_m = \mathbf{T}(\mathcal{E}_m^{100})$ . We perform a simple rounding procedure (see e.g., [316]) on the optimal value of the corresponding optimization problem to obtain a lower bound on (3.76), denoted LB. The resulting upper and lower bounds for fixed  $m$  are averaged over the 250 runs and presented in Table 3.3. The columns ‘Avg. time (s)’ report the average computation time per relaxation in seconds. We observe that optimization over  $\mathbf{T}(\mathcal{E}_m^{100})$  provides significantly stronger bounds than optimization over  $\mathcal{E}_m^{100}$ , for both values of  $m$ . Thus,  $\mathbf{T}(\mathcal{E}_m^n)$  approximates  $\text{CUT}_m^n$  better than the complex elliptope  $\mathcal{E}_m^n$ . To compute bounds over  $\mathbf{T}(\mathcal{E}_m^{100})$ ,  $m \in \{3, 4\}$  we add all triangle facets to  $\mathcal{E}_m^{100}$  at once.

Table 3.3 also reports the total number of (in)equality constraints in the corresponding CSDP in the columns ‘#(in)eq. cons.’. These numbers can be computed as follows. For  $\mathcal{E}_m^n$ , we have  $n$  equality constraints for the unit diagonal, and  $m$  inequalities for each of the  $n(n-1)/2$  upper triangular entries, to ensure  $X_{ij} \in \text{Conv}(\mathcal{U}_m)$ . For  $\mathbf{T}(\mathcal{E}_m^n)$  we have again  $n$  unit diagonal constraints, and  $m$  constraints for each of

the  $n(n-1)/2$  upper triangular entries. Moreover, for each of the  $\binom{n}{3}$  principal  $3 \times 3$  submatrices, we require an additional number of facet defining inequalities. In case  $m = 3$ , we need 18 additional facets that define  $\text{CUT}_3^3$ , according to Theorem 3.17 (note that the other 9 facets are already included to ensure  $X_{ij} \in \text{Conv}(\mathcal{U}_3)$ ). In case  $m = 4$ , we add the 16 ROC equivalent inequalities given by Proposition 3.10, see Remark 3.13, for each of the  $\binom{n}{3}$  principal  $3 \times 3$  submatrices.

		$\mathcal{E}_m^{100}$	$\mathbf{T}(\mathcal{E}_m^{100})$	LB	Avg. time (s)		#(in)eq. cons.	
					$\mathcal{E}_m^{100}$	$\mathbf{T}(\mathcal{E}_m^{100})$	$\mathcal{E}_m^{100}$	$\mathbf{T}(\mathcal{E}_m^{100})$
$m$	3	14337.7	13290.2	9939.7	35.4	173.0	14950	2925550
	4	14849.3	14018.0	11509.3	40.5	169.4	19900	2607100

Table 3.3: Bounds, computation times and number of (in)equality constraints for (3.76), where  $K_m = \mathcal{E}_m^{100}$  or  $K_m = \mathbf{T}(\mathcal{E}_m^{100})$ , and  $m \in \{3, 4\}$ . Results are averaged over 250 runs.

### 3.7.3 MIMO

The multiple-input multiple-output detection problem (MIMO) is a fundamental problem in digital communications. A multiple-input multiple-output channel can be modelled as follows: given a complex channel matrix  $D \in \mathbb{C}^{k \times n}$ , we observe the vector of received signals

$$r := Dc + \sigma v,$$

where  $\sigma > 0$ ,  $c \in \mathcal{U}_m^n$  is the unobserved sent signal and  $v$  is an unobserved vector of noise. The parameter  $\sigma$  governs the so-called signal to noise ratio, see [151]. Observing only  $D$  and  $r$ , MIMO is to retrieve the original signal  $c$ . We refer to e.g., [151, 208, 314] for more details on MIMO. The maximum likelihood estimator (MLE) of  $c$  is

$$\arg \min_{x \in \mathcal{U}_m^n} \|Dx - r\|^2. \quad (3.77)$$

The MLE (3.77) can be approximated by solving instead

$$\min_{X \in K_m} \left\langle \begin{bmatrix} r^H r & -r^H D \\ -D^H r & D^H D \end{bmatrix}, X \right\rangle, \quad (3.78)$$

for  $K_m = \mathcal{E}_m^{n+1}$  or  $K_m = \mathbf{T}(\mathcal{E}_m^{n+1})$ . The complex elliptope  $\mathcal{E}_m^{n+1}$  is defined in (3.3),  $\mathbf{T}(\mathcal{E}_3^{n+1})$  in (3.36), and  $\mathbf{T}(\mathcal{E}_4^{n+1})$  in Section 3.7.2.

We investigate tightness of our new relaxations numerically. We consider  $m \in \{3, 4\}$  and solve (3.78) for different choices of  $K_m$ . Specifically, we set  $n = 99$ , and let  $\sigma \in \{1, 2, 3\}$ . For each combination of  $m$  and  $\sigma$ , we generate 600 matrices  $D \in \mathbb{C}^{109 \times n}$  and vectors  $v \in \mathbb{C}^{109}$ ; these are generated by drawing independent standard complex Gaussians<sup>1</sup>. For each such instance, we solve (3.78) for the different choices of  $K_m$ , and track the rate at which these CSDP relaxations return a (numerical) rank 1

<sup>1</sup>The random variable  $a + bi$  is said to be a standard complex Gaussian if  $a$  and  $b$  are independent, normally distributed random variables with mean 0 and variance 1/2 [173, Definition 24.2.1].

solution. A returned solution matrix is deemed numerically rank 1 if its second largest eigenvalue is strictly smaller than  $10^{-6}$ . If a CSDP relaxation returns a rank 1 solution, the CSDP is said to be tight, since the optimal rank 1 solution can be used to obtain a provably optimal solution to (3.77). Recall that Table 3.3 contains the number of (in)equality constraints for each  $K_m$ .

The results are presented in Table 3.4 for  $m = 3$ , and Table 3.5 for  $m = 4$ . Both tables report the average computation time in seconds, for each relaxation and each  $\sigma$ . We see that adding the facet defining inequalities of  $\text{CUT}_3^3$ , see (3.35), for  $m = 3$  ensures that the CSDP relaxation is tight at a reasonable rate. A similar observation can be made for  $m = 4$ , see Table 3.5. As expected, for increasing values of  $\sigma$ , the CSDP with facet defining inequalities is tight less often. However, without the facet defining inequalities, the CSDP is observed to be tight only once out of the 1200 trials. The computation times for  $\mathbf{T}(\mathcal{E}_m^{100})$  are significantly longer than for  $\mathcal{E}_m^{100}$ . Additionally, we observe that the computation times depend on the rank 1 rates. Indeed, for  $\mathcal{E}_m^{100}$ ,  $m \in \{3, 4\}$ , both the rank 1 rates and the computation times are approximately constant for varying  $\sigma$ . In contrast, for  $\mathbf{T}(\mathcal{E}_m^{100})$ ,  $m \in \{3, 4\}$ , the computation times differ for varying  $\sigma$ . In particular,  $\mathbf{T}(\mathcal{E}_4^{100})$ ,  $\sigma = 3$ , attains the lowest rank 1 rate, and highest average computation time.

$K_m$	$\sigma$			Avg. time (s) per $\sigma$		
	1	2	3	1	2	3
$\mathcal{E}_3^{100}$	0.2%	0.0%	0.0%	56.8	52.7	51.0
$\mathbf{T}(\mathcal{E}_3^{100})$	50.8%	54.5%	51.3%	331.7	320.1	310.5

Table 3.4: Average rate and computation time (over 600 runs) at which (3.78), the CSDP relaxation of MIMO for  $m = 3$  returns a rank 1 solution.

$K_m$	$\sigma$			Avg. time (s) per $\sigma$		
	1	2	3	1	2	3
$\mathcal{E}_4^{100}$	0.0%	0.0%	0.0%	65.0	60.0	59.3
$\mathbf{T}(\mathcal{E}_4^{100})$	49.7%	38.7%	2.5%	271.6	290.5	342.4

Table 3.5: Average rate and computation time (over 600 runs) at which (3.78), the CSDP relaxation of MIMO for  $m = 4$  returns a rank 1 solution.

### 3.7.4 Angular synchronization

In the angular synchronization problem [24, 278], one is given a matrix  $C := cc^H + \sigma W \in \mathbb{C}^{n \times n}$ , where  $c \in \mathcal{U}_\infty^n$  is an unobserved signal,  $\sigma > 0$ , and  $W \in \mathcal{H}^n$  models noise in receiving the signal  $c$ , which one attempts to retrieve. The MLE of  $c$  is given by  $\hat{c} := \arg \max_{x \in \mathcal{U}_\infty^n} x^H C x$  [24, Sect. 2], and  $\hat{c}\hat{c}^H$  can be approximated by

$$\arg \max_{X \in K} \langle C, X \rangle, \quad (3.79)$$

for  $K = \mathcal{E}_\infty^n$ , or some second lifting of  $\text{CUT}_\infty^n$  such as (3.69).

We investigate, for various values of  $\sigma$ , the rate at which the CSDP (3.79) returns a rank 1 solution for different choices of  $K$ . Specifically, we investigate the strength of a parametrized relaxation of  $\text{CUT}_\infty^n$ , induced by basis  $\mathcal{C}_p$ , for  $p \in [0, 1]$ . This basis contains all  $n$  vectors  $\alpha \in \{0, 1\}^{n-1}$  satisfying  $\sum_{i=1}^{n-1} \alpha_i \leq 1$ , plus a fraction  $p$  of vectors from the set  $\{\alpha \in \{0, 1\}^{n-1} : \sum_{i=1}^{n-1} \alpha_i = 2\}$ , chosen uniformly at random (and rounded to nearest integer). The number of elements in this basis can therefore be computed as

$$|\mathcal{C}_p| = n + \left\lceil p \binom{n-1}{2} \right\rceil. \quad (3.80)$$

The induced relaxation of  $\text{CUT}_\infty^n$  is denoted by  $\mathbf{L}^n(\mathcal{C}_p)$ , and defined analogously to (3.69). This relaxation is closely related to the relaxations considered in Section 3.6; note that

$$\text{CUT}_\infty^n \subseteq \mathbf{L}^n(\mathcal{C}_1) = \mathbf{L}^n(\widetilde{\mathcal{A}}) \subseteq \mathbf{L}^n(\mathcal{C}_p) \subseteq \mathbf{L}^n(\mathcal{C}_0) = \mathcal{E}_\infty^n \quad \forall p \in [0, 1], n \in \mathbb{N}.$$

We fix  $n = 25$ ,  $c = \mathbf{1}_n$ , and vary the level of the noise parameter

$$\sigma \in \{(2/3)\sqrt{n}, \sqrt{n}, (4/3)\sqrt{n}\}. \quad (3.81)$$

The chosen levels of  $\sigma$  are in line with [24, Fig. 2], where it is empirically shown that for  $\sigma = (1/3)\sqrt{n}$  and  $K = \mathcal{E}_\infty^n$ , (3.79) very often admits an optimal rank 1 solution. Since we test stronger relaxations than  $\mathcal{E}_\infty^n$ , we have therefore chosen larger values of  $\sigma$ . We generate 100 instances of Hermitian matrices  $W \in \mathcal{H}^n$ , with  $\text{diag}(W) = \mathbf{0}$ , and for which the off-diagonal entries are independently drawn standard complex Gaussians. We track the rate at which the different relaxations, for various values of  $\sigma$ , return rank 1 solutions (with the same zero precision of  $10^{-6}$  as in Section 3.7.3). Results are presented in Table 3.6. There, ‘#cons.’ denotes the number of (complex) equality constraints appearing in the CSDP, and ‘Avg. T. (s)’ stands for the average computation time per relaxation in seconds. Note also that  $|\mathcal{C}_p|$ , see (3.80), denotes the size of the corresponding CSDP, which is equivalent to a real SDP of size  $2|\mathcal{C}_p|$ .

At the tested levels of  $\sigma$ , see (3.81), it can be observed that increasing the relaxation size (i.e.,  $p \rightarrow 1$ ) provides significantly more accurate solutions. For  $p = 1$ , the CSDP is always observed to return a rank 1 solution, and already  $p = 0.75$  offers near-perfect rank 1 rates. The drawback is that the running times significantly increase. However, in practice, if one is interested in computing the MLE of the unobserved signal  $c$ , one should not start by solving the  $\mathbf{L}^n(\mathcal{C}_1)$  or  $\mathbf{L}^n(\mathcal{C}_{0.75})$  relaxation; it is more efficient to solve a smaller relaxation first, say  $\mathbf{L}^n(\mathcal{C}_{0.25})$ , and inspect the optimal solution. If the optimal solution is rank 1, it provides the MLE of  $c$ . If it is not rank 1, one can increase the value of  $p$  and solve the stronger CSDP, continuing so until an optimal rank 1 matrix is observed.

## 3.8 Conclusions

In this chapter we study the complex cut polytope  $\text{CUT}_m^n$ , and its approximations by semidefinite liftings. The considered approximations of  $\text{CUT}_m^n$  are in general not exact, but we investigate under what conditions they are, see Theorem 3.17.

$p$	$ \mathcal{C}_p $	#cons.	Avg. T. (s)	$\sigma/\sqrt{n}$		
				2/3	1	4/3
0	25	25	0.01	18%	4%	1%
0.25	94	612	0.43	61%	34%	27%
0.5	163	1947	3.82	96%	88%	80%
0.75	232	4063	18.49	99%	98%	99%
1	301	6925	71.09	100%	100%	100%

Table 3.6: Rate over 100 runs at which the CSDP relaxation of the angular synchronization problem (3.79), over feasible sets  $\mathbf{L}^n(\mathcal{C}_p)$  with  $n = 25$ , returns a rank 1 solution.

Our first approximation of  $\text{CUT}_m^n$  is the complex elliptope  $\mathcal{E}_m^n$ . This approximation can be strengthened by adding valid inequalities. In Section 3.2 we introduce a framework for numerically comparing valid inequalities, and derive a number of cuts. In Section 3.3 we determine some facet defining inequalities of  $\text{CUT}_3^3$ , and prove that these facets lead to an exact description of  $\text{CUT}_3^3$  (Theorem 3.17). In Section 3.4 we show that a CSDP whose objective function contains only real coefficients, can be equivalently reformulated as a real SDP of the same size (Proposition 3.19).

In Section 3.5, we consider the complex cut polytope  $\text{CUT}_\infty^n$ . We derive several new results for  $n = 4$ , the smallest value for which  $\mathcal{E}_\infty^n$  is not exact (Theorems 3.26 and 3.31). For general  $n$  we provide a method for reducing the size of a second semidefinite lifting without weakening the approximation of  $\text{CUT}_\infty^n$  (Lemma 3.41). In Section 3.6 we investigate the extreme points of  $\mathcal{E}_m^n$  (finite  $m$ ). We find an infinite family of rank 2 extreme points, which proves that the first semidefinite lifting of  $\text{CUT}_m^n$  is never exact (Corollary 3.47).

In Section 3.7 we investigate numerically the value of adding the valid inequalities introduced in Section 3.2 to  $\mathcal{E}_m^n$ ,  $m \in \{3, 4\}$  for CSDPs with randomly generated objectives and the MIMO detection problem. The numerical results show that adding our cuts significantly improves the bounds as well as greatly increases the rate at which the CSDPs return rank 1 solutions. We also test second semidefinite liftings for the angular synchronization problem, and observe that those induce much tighter CSDP relaxations as the size of a basis increases, at the cost of greater computational effort.

For future work, it would be interesting to have Conjecture 3.36 resolved. We are also interested in finding faster methods for solving large CSDPs arising from  $\mathbf{L}^n(\mathcal{C}_p)$ . Table 3.6 shows clearly that larger values of  $p$  greatly improve the strength of relaxations, although the required computational effort (both time and memory) to solve them with off-the-shelf interior-point method solvers increases significantly. A tailored solver might be able to handle much larger values of  $n$  than 25. Many approaches for improving the scalability of real SDP have been proposed in the literature, most of them via exploiting some form of sparsity in the objective function and/or constraints, see e.g., [302, 307, 308, 309] (note that our numerical experiments involved only dense matrices). The authors of [306] remark that some of those approaches can be also applied to CSDPs with only real coefficients. Future research is to investigate how these methods translate for the case of CSDPs over relaxations

of  $\text{CUT}_m^n$ .

# 4 Improved approximation ratios for the quantum max-cut problem on general, triangle-free and bipartite graphs

The quantum max-cut (QMC) problem is to determine the largest energy eigenvalue and corresponding eigenstate of the Hamiltonian

$$H_G := \sum_{\{i,j\} \in E} w_{ij} H_{ij}, \text{ for } H_{ij} := \mathbf{I} - X_i X_j - Y_i Y_j - Z_i Z_j \text{ and } w_{ij} > 0 \quad (4.1)$$

where  $E$  denotes the edge set of a positive edge-weighted graph  $G = (V, E, w)$  on  $n$  vertices, and  $X_i := \mathbf{I}_2^{\otimes(i-1)} \otimes X \otimes \mathbf{I}_2^{\otimes(n-i)}$ . Here  $X$  is one of the well-known Pauli matrices and  $\mathbf{I}_2$  the identity matrix of order two. The matrices  $Y_i$  and  $Z_i$  are similarly defined. We provide further details in Section 5.1.

The QMC problem is an example of a  $k$ -local Hamiltonian problem with  $k = 2$ , since each local Hamiltonian  $H_{ij}$  acts on 2 qubits. The QMC problem is one of the simplest QMA-hard  $k$ -local Hamiltonian problems [255, Thm. 2], as the general  $k$ -local Hamiltonian problem is QMA-hard if and only if  $k \geq 2$  [161]. This motivates the search for polynomial-time (in the number of qubits  $n$ ) approximation algorithms. The QMC problem is the  $k$ -local Hamiltonian problem that has received the most attention when it comes to developing such approximation algorithms, see [112, 145, 153, 156, 164, 190, 191, 246, 247]. This interest is partially due to the fact that the QMC problem is equivalent to the anti-ferromagnetic Heisenberg model from physics.

The study of QMC approximation algorithms is also motivated by the similarities and differences between the QMC problem and the classical NP-hard max-cut problem (see Section 1.3.2). Indeed, taking as local Hamiltonian  $H_{ij} = \mathbf{I} - Z_i Z_j$  reduces problem (4.1) to the max-cut problem. Famously, the max-cut problem admits a polynomial-time 0.878-approximation algorithm [117], which was later shown to be optimal [163] under the Unique Games Conjecture (UGC) [162]. In contrast, the current best-known approximation ratio for the QMC problem is far from its upper bound: the current best-known ratio equals 0.611 [18], while the upper bound equals 0.956 (under UGC and a related conjecture, see [146]). Table 4.1 provides an overview of QMC approximation algorithms for different graph classes, and their approximation ratios. Note that these algorithms are designed to run on classical computers.

A natural question is whether one can design better approximation algorithms if we allow quantum algorithms. Quantum algorithms, based on the well-known Quantum Approximate Optimization Algorithm [84], have recently been proposed in [156, 218]. While these algorithms are empirically shown to achieve strong approximation ratios on some unweighted QMC instances, their approximation ratios are currently unknown. Instead, several asymptotic results have been established when restricting the QMC problem to certain graph classes. For example, let  $|v_p\rangle$  be the output state of the algorithm from [156], with circuit depth  $p \in \mathbb{N}$ . On unweighted bipartite graphs  $G$ , the state  $|v_p\rangle$  converges to an eigenvector corresponding to the largest eigenvalue of  $H_G$ , as  $p \rightarrow \infty$  (the convergence rate is currently not known).

Most QMC approximation algorithms use a semidefinite programming (SDP) relaxation of the QMC problem. SDP relaxations of the QMC problem belong to the field of noncommutative polynomial optimization, and are based on the NPA hierarchy [234, 256], which is the noncommutative variant of the Lasserre hierarchy [175]. The connection to noncommutative polynomials follows from considering the local Hamiltonians in (4.1) as degree-2 polynomials in noncommutative variables  $x_i, y_i$  and  $z_i$  that represent the Pauli matrices (we provide further details in Section 4.2). SDP relaxations of the QMC problem, based on the variables  $x_i, y_i$  and  $z_i$ , were first considered in [112]. Later research [293, 311] investigated SDP relaxations of the QMC problem based on the noncommutative SWAP operators  $S_{ij} := (\mathbf{I} + X_i X_j + Y_i Y_j + Z_i Z_j)/2$ .

**Contributions.** In this chapter, we improve the analysis of the QMC approximation algorithm [191, Alg. 5] for general graphs, and propose two new QMC approximation algorithms, one for triangle-free graphs, and the other for bipartite graphs. We prove that our approximation algorithms for triangle-free and bipartite graphs attain the current best-known approximation ratios for their respective graph classes. Our improved analysis of [191, Alg. 5] has been used in [18], to prove that their QMC algorithm approximation ratio of 0.611, which is the current best-known approximation ratio for the QMC problem on general graphs.

The improved analysis of [191, Alg. 5] follows by showing that a particular vector induced by an SDP relaxation of the QMC problem is contained in the matching polytope. To prove the containment, we compute a bound on the optimal QMC value for all (up to isomorphism) unweighted graphs on  $\leq 13$  vertices. We derive properties of the used SDP relaxation that reduce the required computation time to approximately 16 hours.

Our QMC approximation algorithm for triangle-free graphs is inspired by [164, Alg. 17], which achieves an approximation ratio of 0.582. Our improvements here are due to the use of the matching-based QMC approximation algorithm from [191], and improved parameters. One such parameter is a real-valued function  $\Theta$  that is required to satisfy non-trivial constraints, which we capture using a set of functions  $\mathcal{A}$ . Despite these non-trivial constraints, we prove that our choice of  $\Theta \in \mathcal{A}$  yields an approximation ratio that is at most 0.00009 below the ratio obtained by an optimal  $\Theta \in \mathcal{A}$ .

Our QMC approximation algorithm for bipartite graphs is inspired by [190, Alg. 1], which is suited for general graphs and achieves an approximation ratio of 0.562. One step of [190, Alg. 1] is rounding the used SDP relaxation of the QMC to a cut of the

input graph. For bipartite graphs, we instead take the cut as the bipartition. The resulting algorithm requires a real-valued function  $\Theta \in \mathcal{A}$  as parameter, similar to our algorithm for triangle-free graphs. The function  $\Theta$  maps an optimal solution of the used QMC SDP relaxation, to angles  $\theta_e \in [0, \pi/2]$  used in the outputted state, for all edges  $e$  of the bipartite input graph. To further improve the algorithm, we carefully adjust the value of  $\theta_e$  when the SDP relaxation assigns a sufficiently large objective value to edge  $e$ .

**Outline.** This chapter is organized as follows. Section 4.1 provides preliminaries. In Section 4.2, we state the SDP relaxation that is used for the approximation algorithms, and consider some simple properties of it. Section 4.3 provides the improved analysis of the QMC approximation algorithm from [191]. In Section 4.4 we provide our QMC approximation algorithm for triangle-free graphs, and prove that it achieves an approximation ratio of 0.61383. In Section 4.5, we provide our approximation algorithm for the QMC problem on bipartite graphs, and prove that it achieves an approximation ratio of 0.8162. Concluding remarks and future research directions are given in Section 4.6. Some of our results are based on computations; our code is available in an online repository available at

[https://github.com/LMSinjorgo/QMC\\_proofVerification](https://github.com/LMSinjorgo/QMC_proofVerification).

Reference	Ratio	Remark
[112]	0.498	General graphs, outputs product state
[247]	1/2	General graphs, outputs product state
[145]	0.526	General graphs, uses SOC instead of SDP
[15]	0.531	General graphs
[246]	0.533	General graphs
[190]	0.562	General graphs
[191]	0.595	General graphs
[153]	0.599	General graphs, improved analysis of [191]
Thm. 4.18	0.603	General graphs, improved analysis of [191]
[18]	0.611	General graphs, uses Theorem 4.18
[164]	0.582	Triangle-free graphs
Thm. 4.27	0.61383	Triangle-free graphs
[164]	$\frac{1}{\sqrt{2}} \approx 0.707$	Bipartite graphs
[153]	0.72	Bipartite graphs
[18, 155]	$\frac{1+\sqrt{5}}{4} \approx 0.809$	Bipartite graphs
Thm. 4.33	0.8162	Bipartite graphs

Table 4.1: Lower bounds on the approximation ratios of classical QMC approximation algorithms.

## 4.1 Preliminaries

The  $2 \times 2$  Pauli matrices are given by

$$X := \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, Y := \begin{bmatrix} 0 & -\mathbf{i} \\ \mathbf{i} & 0 \end{bmatrix}, \text{ and } Z := \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, \quad (4.2)$$

where  $\mathbf{i}$  is the imaginary unit. Recall that the matrix  $X_i$  is defined as  $X_i := \mathbf{I}_2^{\otimes(i-1)} \otimes X \otimes \mathbf{I}_2^{\otimes(n-i)}$  where  $n \in \mathbb{N}$ . Matrices  $Y_i$  and  $Z_i$  are similarly defined. The  $3n$  matrices in  $\mathbf{P} := \{X_i, Y_i, Z_i : i \in [n]\}$ , where  $[n] := \{1, \dots, n\}$ , are referred to as extended Pauli matrices, or simply Pauli matrices.

In this chapter we use noncommutative polynomials in the  $3n$  variables  $x_1, \dots, x_n, y_1, \dots, y_n, z_1, \dots, z_n$ . We denote the set of variables by  $\mathbf{p}^n := \{x_i, y_i, z_i : i \in [n]\}$ , and often omit the superscript  $n$  for brevity. A product of noncommutative variables is referred to as a word, or monomial. The length, or degree, of a word is defined as the sum of the exponents in the word. We define  $\langle \mathbf{p}^n \rangle_k$  as the set of all words of degree at most  $k$ , consisting of the  $3n$  variables from  $\mathbf{p}$ . The degree of a polynomial  $f$ , denoted by  $\deg(f)$ , equals the largest degree among its words with nonzero coefficients. We denote the ring of noncommutative polynomials with complex coefficients by  $\mathbb{C}\langle \mathbf{p} \rangle$ , and we let  $\mathbb{C}\langle \mathbf{p} \rangle_k$  be the set of all such polynomials of degree at most  $k$ , where  $k \in \mathbb{N}$ . The sets  $\mathbb{R}\langle \mathbf{p} \rangle$  and  $\mathbb{R}\langle \mathbf{p} \rangle_k$  are defined analogously, using real coefficients instead of complex. The involution  $w \mapsto w^*$  returns the word  $w$  with the order of the symbols reversed. This involution is extended to  $\mathbb{C}\langle \mathbf{p} \rangle$  and  $\mathbb{R}\langle \mathbf{p} \rangle$  by conjugate linearity.

The dual spaces of  $\mathbb{C}\langle \mathbf{p} \rangle$ ,  $\mathbb{C}\langle \mathbf{p} \rangle_k$ ,  $\mathbb{R}\langle \mathbf{p} \rangle$  and  $\mathbb{R}\langle \mathbf{p} \rangle_k$  are denoted by  $\mathbb{C}\langle \mathbf{p} \rangle^*$ ,  $\mathbb{C}\langle \mathbf{p} \rangle_k^*$ ,  $\mathbb{R}\langle \mathbf{p} \rangle^*$  and  $\mathbb{R}\langle \mathbf{p} \rangle_k^*$  respectively. Let  $f_1, \dots, f_m \in \mathbb{C}\langle \mathbf{p} \rangle$ . We define, for  $k \in \mathbb{N}$ ,

$$\langle f_1, \dots, f_m \rangle_k := \left\{ \sum_{i=1}^m g_i f_i h_i : g_i, h_i \in \mathbb{C}\langle \mathbf{p} \rangle, \deg(g_i f_i h_i) \leq k \quad \forall i \in [m] \right\} \quad (4.3)$$

as the two-sided ideal generated by the polynomials  $f_1, \dots, f_m$ , truncated at degree  $k$ . We set  $\langle f_1, \dots, f_m \rangle := \langle f_1, \dots, f_m \rangle_\infty$ . For  $f \in \mathbb{C}\langle \mathbf{p} \rangle$ , we write  $f(\mathbf{P}) \in \mathbb{C}^{2^n \times 2^n}$  to denote the polynomial  $f$  evaluated at the Pauli matrices.

**Definition 4.1** (Set of all graphs on  $n$  vertices). For  $n \in \mathbb{N}$ , we write  $\mathbb{G}_n$  for the set of all simple, undirected, unweighted graphs on  $n$  unlabeled vertices.

**Definition 4.2** (Graph matching). A matching  $\mathcal{M}$  of  $G = (V, E, w)$  is a mapping  $\mathcal{M} : E \rightarrow \{0, 1\}$ , satisfying  $\mathcal{M}_e \mathcal{M}_{e'} = 0$  if edges  $e$  and  $e'$  are adjacent. We say that the edges satisfying  $\mathcal{M}_e = 1$  form a matching. The weight of a matching is given by  $\sum_{e \in E} w_e \mathcal{M}_e$ . We say that a vertex  $i \in V$  is unmatched by  $\mathcal{M}$  if  $\mathcal{M}_e = 0$  for all edges  $e$  incident to  $i$ .

Given an edge-weighted graph  $G = (V, E, w)$ , Edmonds [80] showed that the following linear program (LP), gives the weight of a maximum weight matching in  $G$ :

$$\begin{aligned}
& \max && \sum_{e \in E} w_e x_e \\
& \text{s.t.} && \sum_{j \in N(i)} x_{ij} \leq 1 && \forall i \in V && (4.4a) \\
\text{(MW-LP)} &&& \sum_{e \in E_S} x_e \leq \frac{|S| - 1}{2} && \forall S \subseteq V, |S| \text{ odd} && (4.4b) \\
&&& x_e \geq 0 && \forall e \in E, && (4.4c)
\end{aligned}$$

where  $E_S := \{\{i, j\} \in E : i, j \in S\}$  for all  $S \subseteq V$ , and  $N(i)$  denotes the neighborhood of vertex  $i$ . That is,  $N(i) := \{j \in V : \{i, j\} \in E\}$ .

The feasible set of MW-LP is referred to as the *matching polytope* (of  $G$ ). It is common to refer to the constraints (4.4b) as *odd set inequalities*. Although MW-LP requires an exponential number of such inequalities, it can still be solved in polynomial time using the ellipsoid method [127, Sect. 4]. There also exist algorithms for finding maximum weight matchings that do not use MW-LP, see e.g., [30, 81].

**Definition 4.3** (Edge-induced subgraph). Let  $G = (V, E)$ , and  $\tilde{E} \subseteq E$ . The edge-induced subgraph of  $G$  is denoted by  $G[\tilde{E}]$  and defined as  $G[\tilde{E}] := (\tilde{V}, \tilde{E})$ , for  $\tilde{V} = \{i \in V : \exists j \in V \text{ such that } \{i, j\} \in \tilde{E}\}$ .

**Definition 4.4** (Vertex cover number). For  $G = (V, E)$ , we say that  $S \subseteq V$  is a vertex cover of  $G$  if, for all edges  $\{i, j\} \in E$ , it holds that  $\{i, j\} \cap S \neq \emptyset$ . The vertex cover number of  $G$ , denoted  $\tau(G)$ , is defined as the cardinality of the smallest vertex cover of  $G$ . Equivalently,

$$\tau(G) = \min_{S \subseteq V(G)} \{ |S| : \{i, j\} \cap S \neq \emptyset \ \forall \{i, j\} \in E \}.$$

## 4.2 SDP bounds on $\lambda_{\max}(H_G)$

The matrix  $H_G$ , see (4.1), is of order  $2^n$  which prohibits computing  $\lambda_{\max}(H_G)$  directly. We present an SDP hierarchy, based on noncommutative polynomial optimization [234, 256], see also [112, Sect. 4], that yields upper bounds on  $\lambda_{\max}(H_G)$  in time polynomial in  $n$ .

The problem of computing  $\lambda_{\max}(H_G)$  can be considered as a noncommutative polynomial optimization problem. Indeed, for each of the (extended) Pauli matrices  $X_i, Y_i, Z_i$ ,  $i \in [n]$  we introduce a noncommuting variable  $x_i, y_i, z_i$ , respectively. The (anti)commutation relations of the Pauli matrices can be expressed as  $f(\mathbf{P}) = \mathbf{0}$  for all  $f \in \mathcal{I}$ , where  $\mathcal{I} \subseteq \mathbb{C}\langle \mathbf{p} \rangle$  is the two-sided ideal defined as:

$$\begin{aligned}
\mathcal{I} := \langle & x_i^2 - 1, y_i^2 - 1, z_i^2 - 1, x_i y_j - y_j x_i, x_i z_j - z_j x_i, y_i z_j - z_j y_i, x_i y_i - \mathbf{i} z_i, \\
& x_i z_i + \mathbf{i} y_i, y_i z_i - \mathbf{i} x_i, y_i x_i + \mathbf{i} z_i, z_i x_i - \mathbf{i} y_i, z_i y_i + \mathbf{i} x_i, : i, j \in V, i \neq j \rangle. && (4.5)
\end{aligned}$$

It can be observed that the monomials in

$$\langle \mathbf{p}^n \rangle_k^{\mathcal{I}} := \{u_1 u_2 \cdots u_n : u_i \in \{1, x_i, y_i, z_i\} \ \forall i \in [n], \deg(u_1 \cdots u_n) \leq k\}$$

form a basis of the quotient space  $\mathbb{C}\langle \mathbf{p} \rangle_k / \mathcal{I}$ . We define

$$\mathbb{R}\langle \mathbf{p} \rangle_k^{\mathcal{I}} := \left\{ \sum_{u \in \langle \mathbf{p}^n \rangle_k^{\mathcal{I}}} f_u u : f_u \in \mathbb{R} \quad \forall u \in \langle \mathbf{p}^n \rangle_k^{\mathcal{I}} \right\} \quad (4.6)$$

as the set of real polynomials in the monomials  $\langle \mathbf{p}^n \rangle_k^{\mathcal{I}}$ . Note that any  $f \in \mathbb{R}\langle \mathbf{p} \rangle_k^{\mathcal{I}}$  is symmetric, i.e.,  $f^* \equiv f \pmod{\mathcal{I}}$  and satisfies  $\deg(f) \leq k$ . The set  $\mathbb{R}\langle \mathbf{p} \rangle_k^{\mathcal{I}}$  can be used to define a real SDP relaxation of the QMC problem, with as feasible set

$$F_n^k := \left\{ L \in \mathbb{R}\langle \mathbf{p} \rangle_{2k}^* : \begin{array}{l} L(1) = 1, L(f) = 0 \quad \forall f \in \mathbb{R}\langle \mathbf{p} \rangle_{2k} \text{ satisfying} \\ f + f^* \in \mathcal{I}_{2k}, L(f^2) \geq 0 \quad \forall f \in \mathbb{R}\langle \mathbf{p} \rangle_k^{\mathcal{I}} \end{array} \right\}. \quad (4.7)$$

Note that  $\mathcal{I}_{2k}$  in (4.7) denotes the degree restricted ideal of (4.5), as in (4.3).

**Lemma 4.5.** *Let  $n, k \in \mathbb{N}$ ,  $f \in \mathbb{R}\langle \mathbf{p} \rangle_{2k}$  and  $L \in F_n^k$ . We have that  $L(f) = L(f^*)$ . Additionally, for any  $\tilde{f} \in \mathbb{R}\langle \mathbf{p} \rangle_{2k}$  satisfying  $f \equiv \tilde{f} \pmod{\mathcal{I}}$ ,  $L(f) = L(\tilde{f})$ .*

*Proof.* Consider first the claim  $L(f) = L(\tilde{f})$  whenever  $f \equiv \tilde{f} \pmod{\mathcal{I}}$ . Since  $f \equiv \tilde{f}$ , we have  $f - \tilde{f} \in \mathcal{I}$ , which implies that also  $(f - \tilde{f}) + (f - \tilde{f})^* \in \mathcal{I}$ . Thus,  $L(f - \tilde{f}) = 0$ , which proves the claim by the linearity of  $L$ .

To prove that  $L(f) = L(f^*)$ , write  $f = \sum_{u: \deg(u) \leq 2k} f_u u$  for real numbers  $f_u$  and monomials  $u$ . Any monomial  $u$  satisfies either  $u \equiv u^* \pmod{\mathcal{I}}$ , or  $u \equiv -u^* \pmod{\mathcal{I}}$  [268]. If  $u \equiv u^*$ , then  $L(u) = L(u^*)$  by the previously proved claim. If instead  $u \equiv -u^*$ , then  $u + u^* \in \mathcal{I}_{2k}$ , so that  $L(u) = 0$  by the constraints of  $F_n^k$ . We have

$$L(f) = \sum_{u: \deg(u) \leq 2k} f_u L(u) = \sum_{u: u \equiv u^*} f_u L(u) = \sum_{u: u \equiv u^*} L(f_u u^*) = L(f^*). \quad \square$$

A functional  $L$  that satisfies  $L(f) = L(f^*)$  is said to be symmetric. For symmetric linear functionals, we have that

$$L(f^2) \geq 0 \quad \forall f \in \mathbb{R}\langle \mathbf{p} \rangle_k^{\mathcal{I}} \iff M_k(L) := (L(uv))_{u, v \in \langle \mathbf{p}^n \rangle_k^{\mathcal{I}}} \succeq 0, \quad (4.8)$$

see [45, Lem. 1.44] (and note the equalities  $L(f^2) = L(f^* f)$ ,  $L(uv) = L(u^* v)$ ). The matrix  $M_k(L)$  is referred to as the moment matrix corresponding to  $L$ , and is also known as the noncommutative Hankel matrix [45]. It follows by (4.8) that  $F_n^k$  corresponds to the feasible set of a real SDP.

We define  $H_G(\mathbf{p})$  as the polynomial obtained after replacing the Pauli matrices in (4.1) with their corresponding variables  $x_i, y_i, z_i$ , and  $\mathbf{I}_{2^n}$  with 1. The value

$$\max_{L \in F_n^k} L(H_G(\mathbf{p})) \quad \text{for } k \in \mathbb{N}, \quad (\text{SDP}^k)$$

is an upper bound on  $\lambda_{\max}(H_G)$  that can be computed by solving an SDP. To see that  $\text{SDP}^k$  defines an upper bound on  $\lambda_{\max}(H_G)$ , consider the functional  $\tilde{L}(f) := \frac{1}{2} \langle \psi | (f(\mathbf{P}) + f^*(\mathbf{P})) | \psi \rangle$ , where  $|\psi\rangle$  is a unit length eigenvector of  $H_G$  corresponding to  $\lambda_{\max}(H_G)$ . It can be shown that  $\tilde{L} \in F_n^k$  for any  $k \in \mathbb{N}$ , and achieves an objective value of  $\lambda_{\max}(H_G)$  [190, App. A.1].

The integer  $k$  in  $\text{SDP}^k$  is referred to as the relaxation level of the SDP relaxation. The order of  $M_k(L)$  in (4.8) is given by  $|\langle \mathbf{p}^n \rangle_k^{\mathcal{I}}| = \sum_{i=0}^k 3^i \binom{n}{i} \in \mathcal{O}(n^k)$ . Hence, larger values of  $k$  induce larger SDPs that are harder to solve, but offer tighter upper bounds on  $\lambda_{\max}(H_G)$ . The following result shows that  $\text{SDP}^k$  is exact when  $k = n$ . For a similar statement about an SDP hierarchy based on the SWAP operators, see [311, Thm. 4.10].

**Lemma 4.6.** *Let  $G$  be a graph on  $n$  vertices. The optimal value of  $\text{SDP}^n$  equals  $\lambda_{\max}(H_G)$ .*

*Proof.* Since  $\lambda_{\max}(H_G)\mathbf{I}_{2^n} - H_G \in \mathcal{S}_+^{2^n}$ , there exists a matrix  $M \in \mathcal{S}_+^{2^n}$  that satisfies  $\lambda_{\max}(H_G)\mathbf{I}_{2^n} - H_G = M^2$ , cf. e.g. [144, Thm. 7.2.6]. Define the following set of real matrices

$$\mathbf{P}_{\mathbb{R}}^{\Pi} := \left\{ \bigotimes_{i=1}^n \sigma_i : \sigma_i \in \{\mathbf{I}_2, X, Y, Z\} \quad \forall i \in [n] \right\} \cap \mathbb{R}^{2^n \times 2^n}.$$

Any matrix in  $\mathcal{S}^{2^n}$  can be written as a real linear combination of the matrices in  $\mathbf{P}_{\mathbb{R}}^{\Pi}$  (see Lemma B.5 on Page 211). This implies the existence of a polynomial  $f \in \mathbb{R}\langle \mathbf{p} \rangle_n^{\mathcal{I}}$ , see (4.6), that satisfies  $M = f(\mathbf{P})$ , and so

$$\lambda_{\max}(H_G) - H_G(\mathbf{p}) \equiv f^2 \pmod{\mathcal{I}}. \quad (4.9)$$

Let  $L \in F_n^n$  be an optimal solution of  $\text{SDP}^k$  with  $k = n$ . We have that:

$$\lambda_{\max}(H_G) - L(H_G(\mathbf{p})) = L(\lambda_{\max}(H_G) - H_G(\mathbf{p})) = L(f^2) \geq 0. \quad (4.10)$$

The first equality in (4.10) is due to the linearity of  $L$ , and the fact that  $L(1) = 1$ . The second equality in (4.10) follows from (4.9) and Lemma 4.5. The inequality in (4.10) follows from the definition of  $F_n^n$ .

Rewriting (4.10) yields that  $L(H_G(\mathbf{p})) \leq \lambda_{\max}(H_G)$ . Since it also holds that  $L(H_G(\mathbf{p})) \geq \lambda_{\max}(H_G)$ , we conclude that  $L(H_G(\mathbf{p})) = \lambda_{\max}(H_G)$ , which completes the proof.  $\square$

### 4.3 Improved analysis of a QMC approximation algorithm

In this section, we consider the polynomial-time QMC approximation algorithm from [191], and provide a sharper analysis of this algorithm compared to [191].

To prove the 0.595 approximation ratio of [191, Alg. 5], Lee and Parekh showed how to convert a solution of  $\text{SDP}^k$  for relaxation level  $k = 2$ , to a vector  $h^+ = (h_e^+)_{e \in E}$  (to be defined later) that has the property that  $(\frac{4}{5}h_e^+)_{e \in E}$  is feasible for MW-LP. They did so by showing that  $h^+$  satisfies the odd set inequalities (4.4b) for  $|S| \leq 3$ . Recently, the authors of [153] showed that  $h^+$  satisfies the odd set inequalities for  $|S| \leq 5$ . Here, we show that  $h^+$ , for  $k \geq 2$ , satisfies the odd set inequalities for  $|S| \leq 9$ . This implies that  $(\frac{10}{11}h_e^+)_{e \in E}$  is feasible for MW-LP (see Lemma 4.10). The improved prefactor

10/11 (instead of 4/5) directly translates to an improved approximation ratio of 0.602 for the algorithm. In case  $k \geq 13$ , we can even show that  $(\frac{14}{15}h_e^+)_{e \in E}$  is feasible for MW-LP, which further improves the approximation to 0.603. Our proof of this relies on the exactness of  $\text{SDP}^k$ , with  $k \geq 13$ , for graphs that have at most 13 vertices (see Lemma 4.6).

Let us first present [191, Alg. 5] in Algorithm 1 below. Note that Algorithm 1 is

---

**Algorithm 1:** QMC approximation algorithm [191, Alg. 5]

---

**Input:** Edge-weighted graph  $G = (V, E, w)$ ,  $w > 0$ , on  $n$  vertices. Relaxation level  $k \in \mathbb{N}$ .

- 1 Compute  $\rho_1 = \text{productState}(G, k)$ , see Line 3, and  $\rho_2 = \text{matchingState}(G)$ , see Line 8.
- 2 Output  $\rho_1$  if  $\text{tr}(\rho_1 H_G) \geq \text{tr}(\rho_2 H_G)$ , and  $\rho_2$  otherwise.
- 3 **Function**  $\text{productState}(G, k)$
- 4     Solve  $\text{SDP}^k$  to obtain an optimal moment matrix  $M_k(L)$ . From this  $M_k(L)$ , obtain unit length vectors  $\mathbf{v}(u) \in \mathbb{R}^{|\langle \mathbf{p}^n \rangle_k^{\mathcal{I}}|}$ ,  $u \in \langle \mathbf{p}^n \rangle_k^{\mathcal{I}}$ , that satisfy  $M_k(L)_{u, u'} = \mathbf{v}(u)^\top \mathbf{v}(u')$  for any  $u, u' \in \langle \mathbf{p}^n \rangle_k^{\mathcal{I}}$ . Set
 
$$\mathbf{v}_i := (\mathbf{v}(x_i)^\top, \mathbf{v}(y_i)^\top, \mathbf{v}(z_i)^\top)^\top / \sqrt{3}, \quad i \in V. \quad (4.11)$$
- 5     Sample a random matrix  $R \in \mathbb{R}^{3 \times 3|\langle \mathbf{p}^n \rangle_k^{\mathcal{I}}|}$ , whose elements are independently drawn from the standard normal distribution.
- 6     Compute  $u_i := R\mathbf{v}_i / \|R\mathbf{v}_i\| \in \mathbb{R}^3$  for all  $i \in V$ .
- 7     **return**  $\prod_{i \in V} \frac{1}{2}(\mathbf{I} + u_{i,1}X_i + u_{i,2}Y_i + u_{i,3}Z_i)$
- 8 **Function**  $\text{matchingState}(G)$
- 9     Find a maximum weight matching  $\mathcal{M}$  of  $G$ , see Definition 4.2.
- 10    **return**  $2^{-n} \prod_{\{i,j\}:\mathcal{M}_{\{i,j\}}=1} (\mathbf{I} - X_i X_j - Y_i Y_j - Z_i Z_j)$

---

a polynomial-time algorithm for any fixed  $k \in \mathbb{N}$ . In particular, solving  $\text{SDP}^k$  up to fixed precision, and computing a maximum weight matching [80] are both possible in time polynomial in  $n = |V(G)|$ .

For  $k = 2$ , it is shown [191, Thm. 11] that a lower bound on the approximation ratio of Algorithm 1 is given by the function value  $\alpha(4/5)$  ( $\geq 0.595$ ), for  $\alpha$  given by

$$\alpha(\mu) := \max_{p \in [0,1]} \min_{x \in (-1,1]} \frac{p q(x) + (1-p)(1+3\mu x^+)}{2+2x}, \quad \mu \in [0,1], \quad (4.12)$$

where  $x^+ := \max\{x, 0\}$  and

$$q(x) := 1 + \left( \frac{8+16x}{9\pi} \right) {}_2F_1 \left( \frac{1}{2}, \frac{1}{2}, \frac{5}{2}, \left( \frac{1+2x}{3} \right)^2 \right), \quad (4.13)$$

for  ${}_2F_1$  the hypergeometric function

$${}_2F_1(a, b, c, z) := \sum_{n=0}^{\infty} \frac{(a)_n (b)_n}{(c)_n} \frac{z^n}{n!}, \quad (t)_n := t(t+1) \cdots (t+n-1),$$

with  $(t)_0 = 1$ . Note that  $q(x)$  is well-defined for all  $x \in [-1, 1]$ , in particular for  $x = 1$  due to Gauss's hypergeometric theorem, see e.g., [22, Sect. 1.3].

**Remark 4.7.** The minimization over  $x \in (-1, 1]$  in (4.12) is well-defined, since the minimum does not occur near  $x = -1$ . Indeed, one can verify that  $q(-1) \approx 0.71$ , so that  $\lim_{x \downarrow -1} \frac{pq(x) + (1-p)(1+3\mu x^+)}{2+2x} = +\infty$ , for any  $\mu, p \in [0, 1]$ .  $\triangle$

The variable  $x$  in (4.12) corresponds to  $h_{ij}$ , which is defined as follows:

**Definition 4.8.** Let  $L \in F_n^k$ , see (4.7), with  $n \geq 2$  and  $k \geq 1$ . For  $i, j \in [n]$ ,  $i \neq j$ , define the values

$$h_{ij} := -\frac{1}{2}(1 + L(x_i x_j) + L(y_i y_j) + L(z_i z_j)), \quad h_{ij}^+ := \max\{h_{ij}, 0\}.$$

If  $k \geq 2$ , then  $h_{ij} \in [-1, 1]$ , see [246, Lem. 12], and  $h_{ij}^+ \in [0, 1]$ . Note that  $h_{ij}$  and  $h_{ij}^+$  are functions of  $L \in F_n^k$ . Throughout the rest of this chapter, we implicitly assume this dependence.

Definition 4.8 is similar to [191, Def. 3] with the following difference:  $h := (h_{ij})$  here differs by a factor of 2 compared to the definition of  $h$  in [191]. We note that the lower and upper bound on  $h_{ij}$  are tight for all  $k \geq 2$ . The SDP relaxation  $\text{SDP}^k$  can be expressed in terms of  $h_e$ :

$$\max_{L \in F_n^k} \sum_{e \in E(G)} w_e (2 + 2h_e). \quad (4.14)$$

The approximation ratio of Algorithm 1 is related to the values  $h_e^+$ , as described by the following technical result that is crucial to the remainder of this section.

**Theorem 4.9** ([191]). *Let  $k \in \mathbb{N}$  and let  $G$  be a graph on  $n$  vertices. If  $(\mu h_e^+)_{e \in E(G)}$ , for some  $\mu \in [0, 1]$ , is contained in the matching polytope of  $G$  for all  $L \in F_n^k$ , then the approximation ratio of Algorithm 1 with inputs  $k$  and  $G$  is at least  $\alpha(\mu)$ , where the function  $\alpha$  is defined in (4.12).*

Table 4.2 presents lower bounds on  $\alpha(\mu)$ , for different values of  $\mu$ . It can be observed that  $\alpha(\mu)$  is increasing in  $\mu$ , and we are therefore interested in proving that  $(\mu h_e^+)_{e \in E}$  is contained in the matching polytope for  $\mu$  as close to 1 as possible. To do so, we require the following well-known property of the matching polytope, see e.g., [36, Ex. 6] or [280, App. A].

**Lemma 4.10.** *Let  $x \in \mathbb{R}^{|E|}$  be a vector that satisfies the constraints (4.4a) and (4.4c). If  $x$  also satisfies the odd set inequalities (4.4b) for all  $S \subseteq V$  with  $|S| \leq s$ ,  $s$  odd, then  $\frac{s+1}{s+2}x$  is contained in the matching polytope.*

We will use Lemma 4.10 for  $x = (h_e^+)_{e \in E}$ , see Definition 4.8. Therefore, we need to verify whether the odd set inequalities  $\sum_{e \in E_S} h_e^+ \leq \lfloor s/2 \rfloor$ , with  $s := |S|$  and  $E_S$  defined below MW-LP, hold. To do so, we compute an upper bound on  $\sum_{e \in E_S} h_e^+$  for all feasible solutions to  $\text{SDP}^k$ . That is, we compute

$$\max_{L \in F_n^k} \sum_{1 \leq i < j \leq s} h_{ij}^+ = \max_{z \in \{0,1\}^{\binom{s}{2}}, L \in F_n^k} \sum_{1 \leq i < j \leq s} z_{ij} h_{ij},$$

$\mu$	4/5	6/7	8/9	10/11	12/13	14/15	1
$\alpha(\mu) \geq$	0.595	0.599	0.601	0.602	0.602	0.603	0.606
$p^*$	0.672	0.697	0.709	0.716	0.721	0.724	0.744
$x^*$	0.152	0.153	0.146	0.139	0.142	0.131	0.115

Table 4.2: Lower bounds on  $\alpha(\mu)$  obtained by rounding down  $\alpha(\mu)$  to 3 digits. The rows  $p^*$  and  $x^*$  present a corresponding approximate maximizer/minimizer for the optimization problem in the definition of  $\alpha(\mu)$ .

for odd  $s > 1$  and  $F_s^k$  the feasible set of  $\text{SDP}^k$ , see (4.7). Any fixed  $z \in \{0, 1\}^{\binom{s}{2}}$  defines a graph  $G \in \mathbb{G}_s$ , see Definition 4.1, with edge set  $\{\{i, j\} : z_{ij} = 1\}$ , so that  $\max_{L \in F_s^k} \sum_{1 \leq i < j \leq s} z_{ij} h_{ij} = c(G, k)$ , where

$$c(G, k) := \max_{L \in F_s^k} \sum_{e \in E(G)} h_e. \quad (4.15)$$

Note that  $c(G, k)$  can be computed in polynomial time, up to finite precision, by solving the corresponding SDP. The connection between  $c(G, k)$  and  $\sum_{e \in E} h_e^+$  is clarified by the following result.

**Lemma 4.11.** *Let  $s \geq 2$  and  $k \geq 1$ . For any  $G = (V, E) \in \mathbb{G}_s$  and  $h^+$  as in Definition 4.8, we have*

$$\max_{L \in F_s^k} \sum_{e \in E(G)} h_e^+ \leq \max_{L \in F_s^k} \sum_{1 \leq i < j \leq s} h_{ij}^+ = \max_{G \in \mathbb{G}_s} c(G, k).$$

*Proof.* Since  $h^+ \geq 0$ , we have  $\max_{L \in F_s^k} \sum_{e \in E} h_e^+ \leq \max_{L \in F_s^k} \sum_{1 \leq i < j \leq s} h_{ij}^+$ . Let  $\tilde{L} \in F_s^k$ , see (4.7), be the linear functional that maximizes this upper bound. Let  $\tilde{h}$  be the  $h$  values corresponding to  $\tilde{L}$ , as in Definition 4.8. Consider the graph  $\tilde{G} = (\tilde{V}, \tilde{E})$ , with  $\tilde{V} = [s]$  and  $\tilde{E} := \{\{i, j\} : \tilde{h}_{ij} \geq 0\}$ . Then

$$\max_{L \in F_s^k} \sum_{\{i, j\} \in E} h_{ij}^+ \leq \max_{L \in F_s^k} \sum_{1 \leq i < j \leq s} h_{ij}^+ = c(\tilde{G}, k) \leq \max_{G \in \mathbb{G}_s} c(G, k). \quad (4.16)$$

It remains to show that the second inequality of (4.16) is an equality. To do so, let  $G' \in \arg \max_{G \in \mathbb{G}_s} c(G, k)$  with edge set  $E'$ , and let  $h'$  be the  $h$  values corresponding to an optimal solution of the SDP defining  $c(G', k)$ . By optimality of  $G'$ ,  $h'_{ij} \geq 0$  for all  $\{i, j\} \in E'$ , and  $h'_{ij} \leq 0$  for those  $\{i, j\} \notin E'$ . Hence,

$$\max_{G \in \mathbb{G}_s} c(G, k) = c(G', k) = \sum_{e \in E'} h'_e = \sum_{1 \leq i < j \leq s} \max\{h'_{ij}, 0\} \leq \max_{L \in F_s^k} \sum_{1 \leq i < j \leq s} h_{ij}^+. \quad \square$$

By combining Lemmas 4.10 and 4.11, we obtain the following corollary.

**Corollary 4.12.** *Let  $k, s \in \mathbb{N}$ ,  $s$  odd. Suppose that  $\max_{G \in \mathbb{G}_{s'}} c(G, k) \leq \lfloor s'/2 \rfloor$  for all integers  $s' \leq s$ . Then, for any  $G \in \mathbb{G}_n$  and  $L \in F_n^k$ , the vector  $\left( \frac{s+1}{s+2} h_e^+ \right)_{e \in E(G)}$  is contained in the matching polytope of  $G$ .*

*Proof.* Let  $G = (V, E)$  be a graph. We consider the odd set inequalities (4.4b) for  $x = (h_e^+)_{e \in E(G)}$ . Let  $S \subseteq V$  with  $|S| \leq s$  and  $|S|$  odd. We have, for  $E_S$  as defined below MW-LP, that

$$\sum_{e \in E_S} h_e^+ \leq \sum_{i, j \in S} h_{ij}^+ \leq \max_{G' \in \mathbb{G}_{|S|}} c(G', k) \leq \frac{|S| - 1}{2},$$

by Lemma 4.11 and the assumption of the corollary. Hence,  $(h_e^+)_{e \in E(G)}$  satisfies the odd set inequalities  $\forall S \subseteq V$  with  $|S|$  odd and  $|S| \leq s$ . The claim then follows from Lemma 4.10.  $\square$

Considering Corollary 4.12, we aim to compute  $\max_{G \in \mathbb{G}_s} c(G, k)$  for  $s$  as large as possible. For  $s \in \{3, 4, 5\}$ , the value of  $\max_{G \in \mathbb{G}_s} c(G, k)$  is known.

**Lemma 4.13** ([153, 191]). *For  $k = 2$  and  $s \in \{3, 4, 5\}$ , we have that*

$$\max_{G \in \mathbb{G}_s} c(G, k) = \max_{L \in F_s^k} \sum_{1 \leq i < j \leq s} h_{ij}^+ = \left\lfloor \frac{s}{2} \right\rfloor.$$

In this section, we compute  $\max_{G \in \mathbb{G}_s} c(G, k)$  for  $3 \leq s \leq 13$  and  $k \geq 2$  by exploiting properties of the function  $c(G, k)$  that allow us to reduce the number of computations needed. To prove our results on  $c(G, k)$ , we require an important property of the  $h^+$  and  $h$  values derived from *monogamy of entanglement*. Monogamy of entanglement is a physical property of quantum systems, restricting the maximum entanglement between different quantum states. The connection between monogamy of entanglement and  $\text{SDP}^k$ ,  $k \geq 2$ , is due to [246, Thm. 11] and [191, Lem. 4], which state that

$$\sum_{j \in S} h_{ij} \leq \sum_{j \in S} h_{ij}^+ \leq 1, \quad (4.17)$$

for any fixed  $i$  and subset  $S$  of the vertices with  $i \notin S$ . It is known that (4.17) is not implied by  $\text{SDP}^k$  when  $k \leq 1.5$ , see [246, Thm. 10].

**Remark 4.14.** To compare [246, Thm. 11] with (4.17), observe that  $x_e$  from [246] satisfies  $x_e = (1 + 2h_e)/3$ .  $\triangle$

**Lemma 4.15.** *For any  $G \in \mathbb{G}_s$ ,  $s \geq 2$ , we have the following bounds on  $c(G, k)$ :*

1.  $c(G, k) \leq \sum_{i=1}^p c(G[E^i], k)$ , for  $\{E^1, \dots, E^p\}$  a partition of  $E$ , see Definition 4.3.
2.  $c(G, k) \leq \tau(G)$  for the vertex cover number  $\tau(G)$  as in Definition 4.4 and  $k \geq 2$ .
3.  $c(G, k) \leq s/2$  for  $k \geq 2$ .
4.  $c(G, k) \leq c(G, k - 1)$  for  $k \geq 2$ .
5.  $c(G, k) \leq 1 + \max_{G \in \mathbb{G}_{s-2}} c(G, k)$  if  $G$  contains a vertex of degree 1,  $s \geq 4$  and  $k \geq 2$ .
6.  $c(G, k) \geq \lambda_{\max}(H_G)/2 - |E(G)|$ . This inequality holds with equality if  $k \geq s$ .

*Proof.*

1. For  $\{E^1, \dots, E^p\}$  a partition of  $E(G)$ , we have

$$c(G, k) = \max_{L \in F_s^k} \sum_{e \in E(G)} h_e \leq \sum_{i=1}^p \left( \max_{L \in F_s^k} \sum_{e \in E^i} h_e \right) = \sum_{i=1}^p c(G[E^i], k).$$

The inequality follows from the fact that the maximum of a sum is at most the sum of the maxima, while the equality follows from the definition of  $c(G, k)$ .

2. Let  $V^* \subseteq V$  be a vertex cover of  $G = (V, E)$  satisfying  $|V^*| = \tau(G)$ . For each  $i \in V^*$ , let  $E^i \subseteq E$  be a subset of the edges adjacent to  $i$ , chosen in such a way that the subsets  $E^i$  form a partition of  $E$ . Observe that each edge-induced subgraph  $G[E^i]$  is a star graph. Then, by Item 1 and (4.17), we have  $c(G, k) \leq \sum_{i \in V^*} c(G[E^i], k) \leq |V^*| = \tau(G)$ .

3. The statement follows from (4.17), since

$$\max_{L \in F_s^k} h_{ij}^+ = \max_{L \in F_s^k} \frac{1}{2} \sum_{i \in V} \sum_{j \in N(i)} h_{ij}^+ \leq \frac{s}{2}.$$

The factor  $1/2$  accounts for the fact that we count each edge twice.

4. Let  $L \in F_s^k$ . Let  $\tilde{L} \in \mathbb{R}\langle \mathbf{p} \rangle_{2(k-1)}^*$  be the restriction of  $L$  to inputs from  $\mathbb{R}\langle \mathbf{p} \rangle_{2(k-1)}$ . It follows that  $\tilde{L} \in F_s^{k-1}$ . Moreover,  $\tilde{L}$  and  $L$  induce the same values of  $h$  and  $h^+$ , see Definition 4.8.

5. Without loss of generality, assume that vertex 1 has degree 1, and is connected to vertex 2, i.e.,  $\{1, 2\} \in E$ . Partition  $E := E(G)$  into  $E^1 := \{\{2, i\} : i \in N(2)\}$  and  $E^2 := E \setminus E^1$ . By Item 1, we have  $c(G, k) \leq c(G[E^1], k) + c(G[E^2], k)$ . Since  $G[E^1]$  is a star graph, we have  $c(G[E^1], k) \leq 1$ , see (4.17). Thus,  $c(G, k) \leq 1 + c(G[E^2], k) \leq 1 + \max_{G \in \mathbb{G}_{s-2}} c(G, k)$ , as  $G[E^2]$  is a graph on  $s-2$  vertices.

6. We use (4.14) to obtain

$$c(G, k) = \max_{L \in F_s^k} \sum_{e \in E} h_e = -|E(G)| + \frac{1}{2} \max_{L \in F_s^k} \sum_{e \in E} (2 + 2h_e) \geq \frac{\lambda_{\max}(H_G)}{2} - |E(G)|.$$

The inequality follows from the fact that  $\text{SDP}^k$  provides an upper bound on  $\lambda_{\max}(H_G)$ , which holds with equality if  $k \geq s$ , see Lemma 4.6.  $\square$

**Lemma 4.16.** *Let  $s, k \in \mathbb{N}$ , with  $s, k \geq 2$  and  $s$  odd. Suppose  $\max_{G \in \mathbb{G}_{s'}} c(G, k) = \lfloor s'/2 \rfloor$  for all  $2 \leq s' < s$ . Let  $\mathbf{G} \subseteq \mathbb{G}_s$  be the set of graphs that satisfy the following properties:*

1. *The graph  $G$  is biconnected, triangle-free, and not bipartite.*

2. *For all  $i \in V(G)$ ,  $2 \leq \deg(i) \leq \frac{s-1}{2}$ .*

3. The graph  $G$  has at least  $s$  edges, i.e.,  $|E(G)| \geq s$ .
4. For any nonempty stable set  $S \subseteq V(G)$ ,  $|\cup_{i \in S} N(i)| \geq |S| + 1$ .

We have that

$$\max_{G \in \mathbb{G}_s} c(G, k) = \left\lfloor \frac{s}{2} \right\rfloor \iff \max_{G \in \mathbf{G}} c(G, k) \leq \left\lfloor \frac{s}{2} \right\rfloor. \quad (4.18)$$

*Proof.* See Page 211 in Appendix B.3. □

It can be shown (see Lemma B.8 on Page 213) that any  $G \in \mathbb{G}_s$  that satisfies property 4 of Lemma 4.16, also satisfies  $\deg(i) \geq 2$  for all vertices  $i$ , and  $\tau(G) \geq (s + 1)/2$ .

Finding a list of all graphs in  $\mathbf{G}$  is intractable for large  $s$ , due to the difficulty of checking property 4. However, it is clear that (4.18) remains valid if we replace  $\mathbf{G}$  by a superset of  $\mathbf{G}$  that is simpler to construct. The set of all graphs satisfying properties 1 to 3 is such a superset. The cardinality of this superset is still much smaller compared to  $|\mathbb{G}_s|$ , as we show in Table 4.3. Computing the number of graphs satisfying properties 1 to 3 for  $s = 15$  in Table 4.3 requires approximately 90 minutes. For  $s = 13$ , the computation requires approximately 10 seconds, and less than a second for values of  $s < 13$ . Note that the 5-cycle is the only triangle-free non-bipartite graph on 5 vertices, since non-bipartite graphs must contain an odd cycle.

We will use a relaxed version of property 4 (obtained by constraining  $|S| \leq 2$ ), to reduce the cardinality of the superset even further for the case  $s = 13$  (see Appendix A.7 on Page 201), in order to prove the following result.

$s$	$ \mathbb{G}_s $	#graphs satisfying properties 1 to 3
3	4	0
5	34	1 (the 5-cycle)
7	1044	6
9	274668	219
11	1018997864	26360
13	50502031367952	9035088
15	31426485969804308768	8564316064

Table 4.3: Comparison of  $|\mathbb{G}_s|$  and the number of graphs satisfying properties 1 to 3 of Lemma 4.16.

**Lemma 4.17.** For  $2 \leq s \leq 10$  and  $k \geq 2$ ,  $\max_{G \in \mathbb{G}_s} c(G, k) = \lfloor s/2 \rfloor$ . For  $11 \leq s \leq 14$ ,  $\max_{G \in \mathbb{G}_s} c(G, s) = \lfloor s/2 \rfloor$ .

*Proof.* We use Lemma 4.16 as follows: let  $\mathbf{G}'$  be the set of graphs satisfying properties 1 to 3. Observe that  $\mathbf{G}'$  is a superset of  $\mathbf{G}$ , and that we may replace  $\mathbf{G}$  by  $\mathbf{G}'$  in (4.18). Hence, it remains to show that  $\max_{G \in \mathbf{G}'} c(G, k) \leq \lfloor s/2 \rfloor$ . If  $s$  is even,  $\max_{G \in \mathbf{G}'} c(G, k) \leq \lfloor s/2 \rfloor$ ,  $k \geq 2$ , follows by Item 3 of Lemma 4.15. If  $s \in \{3, 5, 7, 9\}$ , we verify that  $\max_{G \in \mathbf{G}'} c(G, k) \leq \lfloor s/2 \rfloor$ ,  $k = 2$ , by solving the SDPs that define

$c(G, 2)$ . By Item 4 of Lemma 4.15, this also proves the case  $k > 2$ . If  $s \in \{11, 13\}$ , we proceed similarly, except instead of computing  $c(G, s)$  via solving the SDP, we use  $c(G, s) = \lambda_{\max}(H_G)/2 - |E(G)|$ , which is valid due to Item 6 of Lemma 4.15. Computing  $c(G, s)$  in this way requires significantly less time than solving the SDP. We provide more computational details of the proof in Appendix A.7, Page 201.  $\square$

Extending Lemma 4.17 to  $s = 15$  is intractable with current methods: the Hamiltonians corresponding to graphs in  $\mathbb{G}_{15}$  are matrices of order  $2^{15} = 32768$ , and the number of graphs satisfying properties 1 to 3 is of the order  $10^9$ , see Table 4.3. However, Lemma 4.17 already provides the following improved lower bounds on the approximation ratio of Algorithm 1.

**Theorem 4.18.** *For  $k \geq 2$ , the approximation ratio of Algorithm 1 is at least  $\alpha(10/11) \geq 0.602$ , see Table 4.2. If  $k \geq 13$ , the approximation ratio is at least  $\alpha(14/15) \geq 0.603$ .*

*Proof.* By Corollary 4.12 and Lemma 4.17, we have that  $(\frac{10}{11}h_e^+)_{e \in E}$  is contained in the matching polytope when  $k \geq 2$ , and  $(\frac{14}{15}h_e^+)_{e \in E}$  when  $k \geq 13$ . The claim then follows from Theorem 4.9.  $\square$

Our results also imply the following nonlinear inequalities for  $\text{SDP}^k$ .

**Lemma 4.19.** *Let  $L \in F_n^k$ , see (4.7), and  $k \geq 2$ . Then, for the values  $h^+$  derived from  $L$  as in Definition 4.8, we have  $\sum_{1 \leq i < j \leq s} h_{ij}^+ \leq \lfloor s/2 \rfloor$  for all  $s \in \{2, 3, \dots, 10\}$ . These bounds are tight and extend to  $s \in \{11, \dots, 14\}$  if  $k \geq s$ .*

## 4.4 New approximation algorithm on triangle-free graphs

In this section, we propose an approximation algorithm for the QMC problem on triangle-free graphs that achieves an approximation ratio of at least 0.61383. Our algorithm is inspired by [164, Alg. 17], which is also designed for triangle-free graphs and achieves an approximation ratio of at least 0.582.

We present Algorithm 2 on the next page. The input parameter  $\Theta$  of Algorithm 2 is a real-valued function from the set  $\mathcal{A}$  that we will define in Section 4.4.1. The restriction  $\Theta \in \mathcal{A}$  is required for the computation of the approximation ratio of Algorithm 2 in Section 4.4.2.

Algorithm 2 improves over [164, Alg. 17] in three ways. Firstly, we optimize the algorithm parameter  $\Theta$  over a larger space. King chose  $\Theta(x) = Rx^2$  and (numerically) optimized the value of  $R \in \mathbb{R}$  to obtain the highest approximation ratio. In contrast, we consider functions  $\Theta$  over a set  $\mathcal{A}$ , which we prove contains functions of the form  $Rx^2$ , but also  $Rx$  and  $1 - e^{-Rx}$  (Lemma 4.26). We determine a near-optimal  $\Theta \in \mathcal{A}$  in Section 4.4.3. Secondly, in Line 5, we choose the values of  $\varphi_i$  based on a maximum weight matching on a modified graph, which we later show improves over drawing all the  $\varphi_i$  uniformly at random as in [164, Alg. 17]. Thirdly, we output the state  $\text{matchingState}(G)$ , see Algorithm 1, if the state  $|\xi\rangle \langle \xi|$  performs worse.

---

**Algorithm 2:** Approximation algorithm for the QMC problem on triangle-free graphs

---

**Input:** triangle-free, edge-weighted graph  $G = (V, E, w)$ ,  $w > 0$ , on  $n$  vertices. Function  $\Theta \in \mathcal{A}$ , see (4.24).

- 1 Solve  $\text{SDP}^k$  for  $k = 13$  to obtain the vectors  $\mathbf{v}_i$  and values  $(h_e^+)_{e \in E}$ , see (4.11) and Definition 4.8 respectively. Compute the vectors  $u_i \in \mathbb{R}^3$ ,  $i \in V$ , as in Lines 5 and 6 of Algorithm 1 on Page 102. For each  $i \in V$ , let  $\xi_i \in \mathbb{C}^2$  be a unit length vector satisfying  $|\xi_i\rangle \langle \xi_i| = \frac{1}{2} (\mathbf{I}_2 + u_{i,1}X + u_{i,2}Y + u_{i,3}Z)$ .
- 2 Set  $\theta_e := \arcsin \sqrt{\Theta(h_e^+)}$  for all  $e \in E$ .
- 3 For all  $i \in V$ , set  $P_i := \begin{bmatrix} 0 & 1 \\ \exp((2\text{Arg}(\xi_{i,1}^* \xi_{i,2}) + \pi) \mathbf{i}) & 0 \end{bmatrix}$ .
- 4 Compute a maximum weight matching  $\mathcal{M}$  on the modified graph  $\tilde{G} = (V, E, \tilde{w})$ , where  $\tilde{w}$  is defined as

$$\tilde{w}_e := w_e q(h_e) \sqrt{\Theta(h_e^+) (1 - \Theta(1 - h_e^+))}, \quad (4.19)$$

for  $q$  as in (4.13). Let  $U \subseteq V$  be the set of vertices unmatched by  $\mathcal{M}$ .

- 5 For all  $\{i, j\} \in E$ , let

$$\gamma_{ij} := \pi - \text{Arg}(\langle \xi_i | e^{\varphi_j \mathbf{i}} P_j | \xi_j \rangle \langle \xi_j | e^{\varphi_i \mathbf{i}} P_i | \xi_i \rangle), \quad (4.20)$$

where the values of  $\varphi_i$ ,  $i \in V$ , are chosen as follows: for each  $i \in U$ , draw  $\varphi_i \in [0, 2\pi)$  uniformly at random. For  $\{i, j\} \in E$  with  $\mathcal{M}_{\{i,j\}} = 1$ , draw  $\varphi_i \in [0, 2\pi)$  uniformly at random, and choose  $\varphi_j \in [0, 2\pi)$  such that  $\gamma_{ij} = \pi/2$ .

- 6 For all  $i \in V$ , set  $\tilde{P}_i := e^{\varphi_i \mathbf{i}} (\mathbf{I}_2^{\otimes(i-1)} \otimes P_i \otimes \mathbf{I}_2^{\otimes(n-i)})$ .
- 7 Compute  $\rho = \text{matchingState}(G)$ , see Algorithm 1 on Page 102.
- 8 Let

$$|\xi\rangle := \prod_{\{i,j\} \in E} \exp\left(\frac{\mathbf{i}}{2} \text{sgn}(\gamma_{ij}) \theta_{ij} \tilde{P}_i \tilde{P}_j\right) \bigotimes_{i \in V} |\xi_i\rangle. \quad (4.21)$$

- 9 Return the state  $|\xi\rangle \langle \xi|$  if  $\text{tr}(|\xi\rangle \langle \xi| H_G) \geq \text{tr}(\rho H_G)$ , and state  $\rho$  otherwise.
- 

Note that Algorithm 2 computes maximum weight matchings in Lines 4 and 7. To compute the approximation ratio of Algorithm 2, we relate the weight of these matchings to  $h^+$  from Definition 4.8, as in [191]. Given a maximum weight matching  $\mathcal{M}$  on  $G = (V, E, w)$ , this relation is the inequality  $\sum_{e \in E(G)} w_e \mathcal{M}_e \geq \mu \sum_{e \in E(G)} w_e h_e^+$ . Here, the value  $\mu \in [0, 1]$  is such that  $(\mu h_e^+)_{e \in E(G)}$  is contained in the matching polytope. Since Algorithm 2 uses an SDP relaxation level of  $k = 13$ , we may set  $\mu = 14/15$ , as explained in Section 4.3.

### 4.4.1 Properties of $\Theta$

In Algorithm 2, we require that the input function  $\Theta \in \mathcal{A}$ . The set  $\mathcal{A}$  is defined as the set of functions in

$$\mathcal{A}' := \{\Theta : [0, 1] \rightarrow \mathbb{R} : \Theta(0) = 0, \Theta \text{ is increasing, } \Theta(1) \leq 1\}, \quad (4.22)$$

that satisfy, for all  $c \in [0, 1]$ , the equality

$$\min_{x \in [0, c]} (1 - \Theta(x))(1 - \Theta(c - x)) = 1 - \Theta(c). \quad (4.23)$$

That is,

$$\mathcal{A} := \{\Theta \in \mathcal{A}' : \Theta \text{ satisfies (4.23) for all } c \in [0, 1]\}. \quad (4.24)$$

Functions in  $\mathcal{A}$  satisfy the following property, which is a generalization of [164, Cor. 8].

**Lemma 4.20.** *For  $\Theta \in \mathcal{A}$ , and values  $x_0, x_1, \dots, x_p \geq 0$  satisfying  $x_0 + \sum_{s \in [p]} x_s \leq 1$ , we have*

$$\prod_{s \in [p]} (1 - \Theta(x_s)) \geq 1 - \Theta(1 - x_0). \quad (4.25)$$

*Proof.* Since  $\Theta \in \mathcal{A}$ ,  $(1 - \Theta(x_s))(1 - \Theta(x_{s'})) \geq 1 - \Theta(x_s + x_{s'})$  for any distinct  $s, s' \in [p]$ . Iteratively applying the inequality shows that  $\prod_{s \in [p]} (1 - \Theta(x_s)) \geq 1 - \Theta\left(\sum_{s \in [p]} x_s\right)$ . Since  $\Theta$  is an increasing function, and  $\sum_{s \in [p]} x_s \leq 1 - x_0$ , we find  $1 - \Theta\left(\sum_{s \in [p]} x_s\right) \geq 1 - \Theta(1 - x_0)$ .  $\square$

Inequality (4.25) will be used in the next section to compute the approximation ratio of Algorithm 2.

### 4.4.2 Computing the approximation ratio of Algorithm 2

We compute a lower bound on the approximation ratio of Algorithm 2. We use [164, Lem. 12], which provides a lower bound on the expected energy of the state  $|\xi\rangle\langle\xi|$ , see (4.21), in terms of  $\theta_{ij}$ ,  $\gamma'_e$  and  $A_{ij}, B_{ij}$ . Here,

$$\gamma'_e := \gamma_e + \pi \frac{1 - \text{sgn}(\gamma_e)}{2} \in [0, \pi], \quad (4.26)$$

with  $\text{sgn}(0) = 0$ ,  $\gamma_e$  is given by (4.20), and

$$A_{ij} := \prod_{k \in N(i) \setminus \{j\}} \cos \theta_{ik}, \quad B_{ij} := \prod_{k \in N(j) \setminus \{i\}} \cos \theta_{kj}. \quad (4.27)$$

**Lemma 4.21** ([164]). *Let  $G$  be a triangle-free graph used as input to Algorithm 2. Let  $|\xi\rangle$  be as in (4.21), and  $A_{ij}, B_{ij}$  as in (4.27). Then, for any edge  $\{i, j\} \in E(G)$ , we have that*

$$\mathbb{E} \langle \xi | H_{ij} | \xi \rangle \geq \mathbb{E} [E_{ij}] (1 + A_{ij} B_{ij} + \mathbb{E} [\sin \gamma'_{ij}] (A_{ij} + B_{ij}) \sin \theta_{ij}), \quad (4.28)$$

where  $E_{ij} := (1 - u_i^\top u_j)/2$ , see Line 1 of Algorithm 2.

There are two differences in presentation between Lemma 4.21 and [164, Lem. 12]. Firstly, we use a different scaling of  $H_{ij}$  and  $\theta_{ij}$  compared to [164]. Secondly, in [164, Alg. 17], the parameter  $\gamma'_e$  is uniform random on  $[0, \pi]$ . Therefore, in [164, Alg. 17], the expectation of  $\sin \gamma'_e$  is given by  $2/\pi$ . This is in contrast to Algorithm 2, where the distribution of  $\gamma_e$  depends on the matching  $\mathcal{M}$  computed in Line 4 of Algorithm 2. Using that  $\gamma_e$  can be written as

$$\gamma_{ij} = \pi - (\text{Arg}(\langle \xi_i | P_j | \xi_j \rangle \langle \xi_j | P_i | \xi_i \rangle) + \varphi_i + \varphi_j)_{\text{mod } 2\pi},$$

it follows that  $\gamma'_e$ , see (4.26), is uniform random on  $[0, \pi]$  if  $\mathcal{M}_e = 0$ , or equal to  $\pi/2$ , if  $\mathcal{M}_e = 1$ , see Line 5 of Algorithm 2. Thus, we have that

$$\mathbb{E}[\sin \gamma'_e] = (1 - \mathcal{M}_e) \int_0^\pi \frac{\sin x}{\pi} dx + \mathcal{M}_e \sin \frac{\pi}{2} = \frac{2}{\pi} + \frac{\pi - 2}{\pi} \mathcal{M}_e. \quad (4.29)$$

We now define functions that will be used in Theorem 4.22, which establishes a lower bound on the approximation ratio of Algorithm 2. We define

$$f_\Theta(x) := \sqrt{\Theta(x^+)(1 - \Theta(1 - x^+))} \quad (4.30)$$

$$\beta_\Theta(x, \mu) := q(x) \left( 1 - \frac{\Theta(1 - x^+)}{2} + \left( \frac{2}{\pi} + \mu \frac{\pi - 2}{\pi} x^+ \right) f_\Theta(x) \right) \quad (4.31)$$

$$\zeta_\Theta(x, \mu, p) := \frac{p\beta_\Theta(x, \mu) + (1 - p)(1 + 3\mu x^+)}{2 + 2x} \quad (4.32)$$

$$\zeta_\Theta^*(\mu, p) := \min_{x \in (-1, 1]} \zeta_\Theta(x, \mu, p), \quad (4.33)$$

for  $q(x)$  as in (4.13).

**Theorem 4.22.** *The approximation ratio of Algorithm 2 for triangle-free graphs is at least*

$$\max_{p \in [0, 1]} \zeta_\Theta^*(14/15, p).$$

*Proof.* Let  $e = \{i, j\}$ . Using Lemma 4.20, we find, for  $A_{ij}$  as in (4.27),

$$A_{ij} = \prod_{k \in N(i) \setminus \{j\}} \cos \theta_{ik} = \prod_{k \in N(i) \setminus \{j\}} \sqrt{1 - \Theta(h_{ik}^+)} \geq \sqrt{1 - \Theta(1 - h_e^+)}, \quad (4.34)$$

and similarly  $B_{ij} \geq (1 - \Theta(1 - h_e^+))^{1/2}$ . Indeed, Lemma 4.20 applies here since the values  $(h_{ik}^+)_{k \in N(i)}$  satisfy  $h_{ik}^+ \geq 0$  for all  $k \in N(i)$ , and  $h_{ij}^+ + \sum_{k \in N(i) \setminus \{j\}} h_{ik}^+ \leq 1$ , due to (4.17).

We substitute (4.34) into (4.28) (note that  $\sin \theta_e = \sqrt{\Theta(h_e^+)}$ ). Additionally, we substitute  $\mathbb{E}[E_{ij}] = q(h_e)/2$  [37, Lem. 2.1], for  $q$  as in (4.13) (see also [164, Lem. 13]). This yields the following, where  $f_\Theta$  is as in (4.30):

$$\begin{aligned} \mathbb{E}[\xi | H_e | \xi] &\geq \frac{q(h_e)}{2} (1 + (1 - \Theta(1 - h_e^+)) + 2 \mathbb{E}[\sin \gamma'_e] f_\Theta(h_e)) \\ &= q(h_e) \left( 1 - \frac{\Theta(1 - h_e^+)}{2} + \left( \frac{2}{\pi} + \frac{\pi - 2}{\pi} \mathcal{M}_e \right) f_\Theta(h_e) \right) \\ &= \beta_\Theta(h_e, 0) + \frac{\pi - 2}{\pi} \frac{\tilde{w}_e}{w_e} \mathcal{M}_e. \end{aligned} \quad (4.35)$$

The first equality in (4.35) is due to (4.29), for  $\mathcal{M}$  the matching computed in Line 4 of Algorithm 2. The second equality in (4.35) is due to the definitions of  $\tilde{w}$  and  $\beta_\Theta$ , see (4.19) and (4.31) respectively. By combining (4.1) and (4.35), we have that

$$\mathbb{E} \langle \xi | H_G | \xi \rangle = \sum_{e \in E} w_e \mathbb{E} \langle \xi | H_e | \xi \rangle \geq \sum_{e \in E} \left( w_e \beta_\Theta(h_e, 0) + \frac{\pi - 2}{\pi} \tilde{w}_e \mathcal{M}_e \right). \quad (4.36)$$

Note that  $\mathcal{M}$  corresponds to a maximum weight matching on the graph with positive edge weights  $\tilde{w}$ . By Corollary 4.12 and Lemma 4.17,  $(\mu h_e^+)_{e \in E}$  is contained in the matching polytope for  $\mu = 14/15$ . Therefore,

$$\sum_{e \in E} \tilde{w}_e \mathcal{M}_e \geq \mu \sum_{e \in E} \tilde{w}_e h_e^+, \quad (4.37)$$

for  $\tilde{w}_e = w_e q(h_e) f_\Theta(h_e)$ , see (4.19). We substitute (4.37) in (4.36) to obtain

$$\begin{aligned} \mathbb{E} \langle \xi | H_G | \xi \rangle &\geq \sum_{e \in E} \left( w_e \beta_\Theta(h_e, 0) + \mu \frac{\pi - 2}{\pi} \tilde{w}_e h_e^+ \right) \\ &= \sum_{e \in E} w_e q(h_e) \left[ 1 - \frac{\Theta(1 - h_e^+)}{2} + \frac{2}{\pi} f_\Theta(h_e) + \mu \frac{\pi - 2}{\pi} h_e^+ f_\Theta(h_e) \right] \\ &= \sum_{e \in E} w_e \beta_\Theta(h_e, \mu). \end{aligned} \quad (4.38)$$

Here, we have also used the definition of  $\beta_\Theta$ , see (4.31). As for  $\rho$ , defined in Line 7 of Algorithm 2, one can show that

$$\text{tr}(\rho H_G) \geq \sum_{e \in E} w_e (1 + 3\mu h_e^+),$$

see [191, Eq. 9]. The expected energy attained by Algorithm 2 satisfies

$$\begin{aligned} \mathbb{E} \max \{ \langle \xi | H_G | \xi \rangle, \text{tr}(\rho H_G) \} &\geq \max_{p \in [0, 1]} [p \mathbb{E} \langle \xi | H_G | \xi \rangle + (1 - p) \text{tr}(\rho H_G)] \\ &\geq \max_{p \in [0, 1]} \sum_{e \in E} w_e (p \beta_\Theta(h_e, \mu) + (1 - p) (1 + 3\mu h_e^+)) \\ &\geq \max_{p \in [0, 1]} \zeta_\Theta^*(\mu, p) \sum_{e \in E} w_e (2 + 2h_e) \geq \max_{p \in [0, 1]} \zeta_\Theta^*(\mu, p) \lambda_{\max}(H_G). \end{aligned} \quad (4.39)$$

We have used (4.38) for the second inequality in (4.39). The fourth inequality in (4.39) follows from the fact that  $\text{SDP}^k$  provides an upper bound on  $\lambda_{\max}(H_G)$ , as explained in Section 4.2. Note that  $h_e \in [-1, 1]$ , see Definition 4.8, while the minimization in  $\zeta_\Theta^*(\mu, p)$  is done over  $x \in (-1, 1]$ . This distinction is made to avoid division by 0 and can be done without loss of generality, see Remark 4.7.  $\square$

It remains to find a function  $\Theta \in \mathcal{A}$  for which  $\max_{p \in [0, 1]} \zeta_\Theta^*(14/15, p)$  is high.

### 4.4.3 Finding candidate functions $\Theta \in \mathcal{A}$

The set  $\mathcal{A}$ , see (4.24), is difficult to characterize in general. Here, we derive a sufficient condition for  $\Theta \in \mathcal{A}$ . This condition can be stated in terms of logarithmically concave functions, which are defined as follows (see also [33, Sect. 3.5]). Note that we define  $\log 0 := -\infty$ .

**Definition 4.23.** A nonnegative real-valued function  $f$ , with convex domain, is logarithmically concave (log-concave for short) if and only if  $\log f$  is concave (with  $\log 0 = -\infty$ ).

Equivalently,  $f$  is log-concave if and only if

$$f(wx + (1-w)y) \geq f(x)^w f(y)^{1-w} \quad \forall w \in [0, 1] \quad (4.40)$$

and for all  $x$  and  $y$  in the domain of  $f$ . It can also be shown that products of log-concave functions are log-concave [33, Sect. 3.5.2].

**Lemma 4.24.** Let  $\Theta \in \mathcal{A}'$ , see (4.22). If  $1 - \Theta(x)$  is log-concave, then  $\Theta \in \mathcal{A}$ .

*Proof.* Let  $c \in [0, 1]$  and define  $f(x) := (1 - \Theta(x))(1 - \Theta(c - x))$  for  $x \in [0, c]$ . We need to show that  $\min_{x \in [0, c]} f(x) = f(0) = f(c)$ . The case  $c = 0$  is trivial, so we assume  $c \in (0, 1]$ . Observe that log-concavity of  $1 - \Theta(x)$  implies log-concavity of  $1 - \Theta(c - x)$ , for  $x \in [0, c]$ . Since  $f$  is then the product of log-concave functions,  $f$  is also log-concave with domain  $[0, c]$ . It follows from (4.40) and  $f(0) = f(c)$  that for all  $x \in [0, c]$ ,

$$f(x) = f\left(c \frac{x}{c}\right) \geq f(0)^{1-x/c} f(c)^{x/c} = f(0),$$

which completes the proof.  $\square$

Lemma 4.24 implies the following weaker, but simpler, result.

**Lemma 4.25.** Let  $\Theta \in \mathcal{A}'$ , see (4.22). If  $\Theta$  is convex, then  $\Theta \in \mathcal{A}$ .

*Proof.* Since  $\Theta$  is convex and  $\Theta(x) \leq 1$ ,  $1 - \Theta(x)$  is a nonnegative concave function. Nonnegative concave functions are log-concave [33, Sect. 3.5.1], which implies by Lemma 4.24 that  $\Theta \in \mathcal{A}$ .  $\square$

The following result follows directly from Lemma 4.25

**Lemma 4.26.** The functions  $1 - e^{-R_1 x}$ ,  $R_1 \geq 0$ , and  $R_2 x^c$ ,  $R_2 \in [0, 1]$ ,  $c \geq 1$ , are contained in  $\mathcal{A}$ .

*Proof.* It can be verified that the functions are contained in  $\mathcal{A}'$ , see (4.22). Since the functions are also convex, the result follows from Lemma 4.25.  $\square$

We choose  $\Theta(x) = 1 - e^{-x/20}$ , which is an element of  $\mathcal{A}$  by Lemma 4.26. It follows by Theorem 4.22 that Algorithm 2, with this choice of  $\Theta$ , achieves an approximation ratio of at least

$$\max_{p \in [0, 1]} \zeta_{1-e^{-x/20}}^*(14/15, p) \geq \zeta_{1-e^{-x/20}}^*(14/15, p^*) = 0.61383, \quad (4.41)$$

where  $p^* = (2 \cdot 0.61383 - 1) / (\beta_{1-e^{-x/20}}(0, 14/15) - 1)$ . This value of  $p^*$  is chosen such that  $\zeta_{\Theta}(0, 14/15, p^*) = 0.61383$ , see (4.32). Consider the function  $\zeta(x) := \zeta_{1-e^{-x/20}}(x, 14/15, p^*)$ , plotted in Figure 4.1. We briefly elaborate on how one can prove the statement  $\zeta_{1-e^{-x/20}}^*(14/15, p^*) = \min_{x \in (-1, 1]} \zeta(x) = 0.61383$  in (4.41). It can be shown (details omitted) that  $\zeta$  is decreasing for  $x \in (-1, 0]$  and increasing for  $x \in [0, 0.035]$ , so that  $\min_{x \in (-1, 0.035]} \zeta(x) = \zeta(0) = 0.61383$ . For  $x \in [0.035, 1]$ , it is possible to derive a lower bound on  $\zeta'(x)$ , and evaluate  $\zeta$  on a fine grid in the interval  $[0.035, 1]$ . By combining the lower bound on  $\zeta'$  with the mean value theorem, one can prove that  $\zeta(x) \geq 0.61383$  for any  $x \in [0.035, 1]$ .

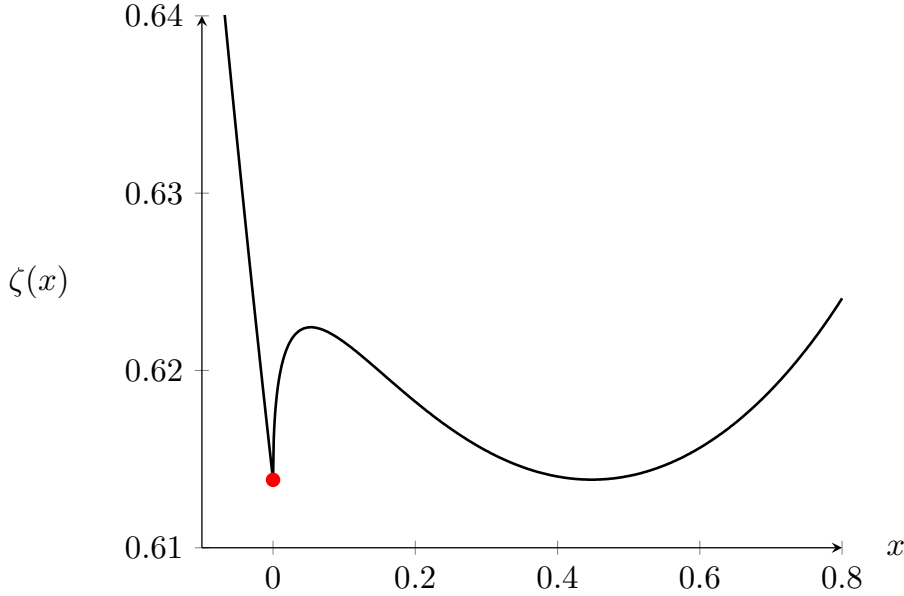


Figure 4.1: Plot of  $\zeta(x) := \zeta_{\Theta}(x, 14/15, p^*)$  for  $x \in [-0.1, 0.8]$ ,  $\Theta(x) = 1 - e^{-x/20}$  and  $p^*$  as in (4.41). We have  $\min_{x \in (-1, 1]} \zeta(x) = \zeta(0) = 0.61383$ , marked by the dot.

The approximation ratio in (4.41) is greater than  $\alpha(1) = 0.606$  (see Table 4.2), which is a lower bound on the approximation ratio of Algorithm 1 if  $(h_e^+)_{e \in E}$  is contained in the matching polytope, see Theorem 4.9. The optimal approximation ratio of Algorithm 2, with respect to  $\Theta \in \mathcal{A}$ , is given by  $\max_{p \in [0, 1], \Theta \in \mathcal{A}} \zeta_{\Theta}^*(\mu, p)$ . Equation (4.41) proves that  $\max_{p \in [0, 1], \Theta \in \mathcal{A}} \zeta_{\Theta}^*(\mu, p) \geq 0.61383$ . Let us now provide an upper bound on this maximum.

**Theorem 4.27.** *The best provable lower bound on the approximation ratio of Algorithm 2, given by  $\max_{p \in [0, 1], \Theta \in \mathcal{A}} \zeta_{\Theta}^*(14/15, p)$ , satisfies*

$$0.61383 \leq \max_{p \in [0, 1], \Theta \in \mathcal{A}} \zeta_{\Theta}^*(14/15, p) < 0.61392, \quad (4.42)$$

for  $\mathcal{A}$  as in (4.24) and  $\zeta_{\Theta}^*$  as in (4.33).

*Proof.* Let  $\mu = 14/15$ . The lower bound in (4.42) is due to (4.41). For the upper bound, observe that by the definition of  $\zeta_{\Theta}^*(\mu, p)$ , see (4.33), we have that

$$\zeta_{\Theta}^*(\mu, p) \leq \min \{ \zeta_{\Theta}(0, \mu, p), \zeta_{\Theta}(1/2, \mu, p) \}. \quad (4.43)$$

Since  $\max_{p \in [0,1], \Theta \in \mathcal{A}} \zeta_{\Theta}^*(\mu, p) \geq 0.61383$ , we may consider only  $\Theta \in \mathcal{A}$  and  $p \in [0, 1]$  satisfying  $\zeta_{\Theta}(0, \mu, p) \geq 0.61383$ . Solving

$$\zeta_{\Theta}(0, \mu, p) = \frac{pq(0)(1 - \Theta(1)/2) + 1 - p}{2}$$

for  $\Theta(1)$  yields

$$\Theta(1) = 2 - \frac{2}{q(0)} - \frac{4\zeta_{\Theta}(0, \mu, p) - 2}{pq(0)}. \quad (4.44)$$

Since  $\Theta(1) \geq 0$ , (4.44) implies that

$$p \geq \frac{2\zeta_{\Theta}(0, \mu, p) - 1}{q(0) - 1}. \quad (4.45)$$

We have  $\zeta_{\Theta}(1/2, \mu, p) = f(\Theta(1/2), p)$ , where

$$f(z, p) := \frac{1}{3} \left( pq(1/2) \left( 1 - \frac{z}{2} + C_1 \sqrt{z(1-z)} \right) + (1-p)C_2 \right), \quad (4.46)$$

and  $C_1 := (4 + \mu(\pi - 2))/(2\pi)$ ,  $C_2 := (1 + 3\mu/2)$ .

By the properties of functions in  $\mathcal{A}$ , see (4.24), we have

$$1 - \Theta(1) = \min_{x \in [0,1]} (1 - \Theta(x))(1 - \Theta(1-x)) \leq (1 - \Theta(1/2))^2,$$

from where it follows that  $\Theta(1/2) \leq 1 - \sqrt{1 - \Theta(1)}$  (the expression  $\sqrt{1 - \Theta(1)}$  is well-defined since  $\Theta(1) \leq 1$ ). By substituting (4.44) for  $\Theta(1)$  in this inequality, we find that  $\Theta(1/2) \leq 1 - \sqrt{1 - \Theta(1)}$  is equivalent to  $\Theta(1/2) \leq h(\zeta_{\Theta}(0, \mu, p), p)$ , where

$$h(z, p) := 1 - \sqrt{\frac{4z-2}{pq(0)} + \frac{2}{q(0)} - 1}. \quad (4.47)$$

For future reference, we observe the following two properties of the function  $h$ :

$$\frac{\partial h(z, p)}{\partial z} < 0 \quad \text{and} \quad \frac{\partial h(z, p)}{\partial p} > 0 \quad \text{if} \quad z > \frac{1}{2}. \quad (4.48)$$

We claim that for  $f$  as in (4.46), we have

$$f(\Theta(1/2), p) \leq f(h(\zeta_{\Theta}(0, \mu, p), p), p). \quad (4.49)$$

As  $\Theta(1/2) \leq h(\zeta_{\Theta}(0, \mu, p), p)$ , claim (4.49) follows by showing that  $f(z, p)$  is increasing in  $z$  on the interval  $0 \leq z \leq h(\zeta_{\Theta}(0, \mu, p), p)$  and  $p \in [0, 1]$ . Indeed, one can verify that for  $p \in [0, 1]$  and  $z \in \left(0, \frac{1}{2} \left(1 - \frac{1}{\sqrt{4C_1^2 + 1}}\right)\right)$ ,

$$\frac{\partial f(z, p)}{\partial z} = \frac{pq(1/2)}{6} \left( \frac{C_1(1-2z)}{\sqrt{z(1-z)}} - 1 \right) \geq 0. \quad (4.50)$$

The interval of  $z$  can be found by solving  $\partial f(z, p)/\partial z = 0$ . To prove (4.49), it remains to show that  $h(\zeta_{\Theta}(0, \mu, p), p) \leq \frac{1}{2} \left(1 - \frac{1}{\sqrt{4C_1^2 + 1}}\right)$ . Recall that  $\zeta_{\Theta}(0, \mu, p) \geq 0.61383$ , which we use to obtain

$$h(\zeta_{\Theta}(0, \mu, p), p) \leq h(0.61383, p) \leq h(0.61383, 1) \leq \frac{1}{2} \left(1 - \frac{1}{\sqrt{4C_1^2 + 1}}\right), \quad (4.51)$$

where the first and second inequality in (4.51) follow from (4.48). The last inequality in (4.51) follows by computing the two numbers. Hence, claim (4.49) follows.

Combining (4.43),  $\zeta_{\Theta}(1/2, \mu, p) = f(\Theta(1/2), p)$  and (4.49) yields

$$\begin{aligned} \zeta_{\Theta}^*(\mu, p) &\leq \min \{ \zeta_{\Theta}(0, \mu, p), f(h(\zeta_{\Theta}(0, \mu, p), p), p) \} \\ &\leq \max_{z \in [0.61383, 1]} \min \left\{ z, \max_{p \in \left[\frac{2z-1}{q(0)-1}, 1\right]} f(h(z, p), p) \right\}. \end{aligned} \quad (4.52)$$

In (4.52), the constraint on  $z$  is due to our assumption  $0.61383 \leq \zeta_{\Theta}(0, \mu, p) = z$ , whereas the constraint on  $p$  is due to (4.45). We define  $r := 0.61392$ , and show that (4.52) is upper bounded by  $r$ . To solve the maximization problem in  $z$  in (4.52), we first note that when  $z < r$  we have  $\zeta_{\Theta}^*(\mu, p) \leq z < r$ . Thus, we may restrict to  $z \in [r, 1]$ . We proceed by proving that

$$\max_{z \in [r, 1], p \in \left[\frac{2z-1}{q(0)-1}, 1\right]} f(h(z, p), p) < r, \quad (4.53)$$

which would prove  $\zeta_{\Theta}^*(\mu, p) \leq r$  by (4.52). To simplify (4.53), we claim that

$$f(h(z, p), p) \leq f(h(r, p), p) \quad (4.54)$$

for all  $z \in [r, 1]$  and  $p \in [0, 1]$ . We showed in (4.50) that  $\partial f(z, p)/\partial z \geq 0$  for all  $0 < z \leq \frac{1}{2} \left(1 - \frac{1}{\sqrt{4C_1^2 + 1}}\right)$ , so that claim (4.54) follows by showing that  $0 \leq h(z, p) \leq h(r, p) \leq \frac{1}{2} \left(1 - \frac{1}{\sqrt{4C_1^2 + 1}}\right)$ . To prove these inequalities, we use  $z \geq r > 1/2$  and (4.48) to obtain  $h(z, p) \leq h(r, p) \leq h(r, 1) \leq \frac{1}{2} \left(1 - \frac{1}{\sqrt{4C_1^2 + 1}}\right)$ . Here, the last inequality can be verified by computing the two numbers. Hence, claim (4.54) is proven.

Note that the interval  $\left[\frac{2z-1}{q(0)-1}, 1\right]$  with  $z \in [r, 1]$ , is largest for  $z = r$ . Combining this observation with (4.54) yields that

$$\max_{z \in [r, 1], p \in \left[\frac{2z-1}{q(0)-1}, 1\right]} f(h(z, p), p) \leq \max_{p \in \left[\frac{2r-1}{q(0)-1}, 1\right]} f(h(r, p), p). \quad (4.55)$$

Lemma B.9, Page 213, proves that  $\max_{p \in \left[\frac{2r-1}{q(0)-1}, 1\right]} f(h(r, p), p) < r$ . Combined with (4.52) and (4.55), we find that  $\zeta_{\Theta}^*(\mu, p) < r = 0.61392$ , which finishes the proof.  $\square$

## 4.5 New approx. algorithm on bipartite graphs

In this section we consider an approximation algorithm for the QMC problem on bipartite graphs that achieves an approximation ratio of 0.8162. The QMC problem, restricted to bipartite graphs, belongs to the complexity class StoqMA [63, Obs. 30] and remains 'notoriously difficult to solve' [112, Sect. 4]. It is known that on bipartite graphs, the QMC problem is equivalent to the *Einstein, Podolsky, Rosen* (EPR) problem, which is another 2-local Hamiltonian problem introduced in [164, Problem 2] (specifically, the Hamiltonian corresponding to the EPR problem is unitarily similar to  $H_G$ ). In the same paper, King provides a classical approximation algorithm for the EPR problem that achieves an approximation ratio of  $1/\sqrt{2}$  ( $\approx 0.707$ ).

---

### Algorithm 3: QMC approximation algorithm for bipartite graphs

---

- Input:** bipartite graph  $G = (V = V^0 \cup V^1, E, w)$ , function  $\Theta \in \mathcal{A}$ , see (4.24), real numbers  $h_{\max} \in [\sqrt{3}/2, 1]$  and  $\theta^* \in [0, 1]$ .
- 1 For all  $i \in V$ , set  $|z_i\rangle = |0\rangle$  if  $i \in V^0$ , and  $|z_i\rangle = |1\rangle$  if  $i \in V^1$ .
  - 2 Solve  $\text{SDP}^k$  for  $k = 2$  to obtain the values  $(h_e)_{e \in E}$ , see Definition 4.8, and for all  $e \in E$ , set

$$\theta_e := \begin{cases} \arcsin \sqrt{\Theta(h_e^+)} & \text{if } h_e \leq h_{\max} \\ \arcsin \sqrt{\theta^*} & \text{else.} \end{cases} \quad (4.56)$$

- 3 Output the state  $|\phi\rangle \langle \phi|$ , where

$$|\phi\rangle := \prod_{\{i,j\} \in E: i \in V^0, j \in V^1} \exp\left(\frac{\mathbf{i}}{2} \theta_{ij} Y_i X_j\right) \bigotimes_{i \in V} |z_i\rangle. \quad (4.57)$$


---

The algorithm we consider is Algorithm 3, which is inspired by [190, Alg. 1] from Lee. Lee's algorithm is suited for general graphs, achieves an approximation ratio of 0.562, and differs from Algorithm 3 in Lines 1 and 2. In Line 1, both algorithms determine the value of  $|z_i\rangle \in \{|0\rangle, |1\rangle\}$  based on a partition of the vertex set  $V$ . For Algorithm 3, this partition is already given as  $\{V^0, V^1\}$ , i.e., the bipartition of  $G$ . In contrast, [190, Alg. 1] partitions  $V$  as  $\{V', V \setminus V'\}$ , where  $V'$  is constructed as follows: pick a variable  $u \in \{x, y, z\}$  uniformly at random. Consider the vectors  $(\mathbf{v}(u_i))_{i \in V}$ , where  $\mathbf{v}(u_i)$  is as in (4.11) (for example, if  $u = x$ , then  $\mathbf{v}(u_i) = \mathbf{v}(x_i)$ ). Draw a vector  $r$  uniformly from the unit sphere of appropriate dimension. Let  $V' := \{i \in V : \text{sgn}(\mathbf{v}(u_i)^\top r) = 1\}$ . The random partition  $\{V', V \setminus V'\}$  is not guaranteed to recover the bipartition of a bipartite graph. That is, when given a bipartite graph as input, it is not guaranteed that the vertices in  $V'$  are pairwise non-adjacent.

In Line 2, both algorithms set  $\theta_e$  as a function of  $h_e^+$ . Lee chose  $\Theta(x) = 1 - e^{-Rx} \in \mathcal{A}$  (see Lemma 4.26) and (numerically) optimized the value of  $R \geq 0$  to obtain the highest approximation ratio. In contrast, we choose  $\Theta(x) = Rx$  in Theorem 4.33, which is also contained in  $\mathcal{A}$  by Lemma 4.26. Additionally, in Theorem 4.34 we

provide an upper bound on the approximation ratio attained by the best possible  $\Theta \in \mathcal{A}$ . Lastly, Lee's algorithm does not use  $h_{\max}$ , see (4.56), which allows for adjusting the value of  $\theta_e$  when  $h_e$  is sufficiently large. These differences in Lines 1 and 2 ensure that Algorithm 3 achieves a higher approximation ratio than [190, Alg. 1] on bipartite graphs. To compute this approximation ratio in terms of  $\Theta$ ,  $h_{\max}$  and  $\theta^*$ , we require some preparatory results and definitions. We present first [190, Lem. 11].

**Lemma 4.28** ([190]). *Let  $|\phi\rangle$  be as in (4.57) and  $|z_i\rangle, |z_j\rangle$  as in Line 1 of Algorithm 3. For any  $i, j \in V$ ,  $\langle \phi | H_{ij} | \phi \rangle \geq 0$ . If  $|z_i\rangle \neq |z_j\rangle$ , then, for  $A_{ij}$  and  $B_{ij}$  as in (4.27),*

$$\langle \phi | H_{ij} | \phi \rangle \geq 1 + A_{ij}B_{ij} + (A_{ij} + B_{ij}) \sin \theta_{ij}. \quad (4.58)$$

The small discrepancy between our Lemma 4.28 and [190, Lem. 11] is due to the different scaling of  $\theta$ . If  $|z_i\rangle \neq |z_j\rangle$  for some edge  $\{i, j\}$ , we say that edge  $\{i, j\}$  is cut. If  $\{i, j\}$  is a cut edge,  $\langle \phi | H_{ij} | \phi \rangle$  satisfies (4.58), which is a stronger lower bound than  $\langle \phi | H_{ij} | \phi \rangle \geq 0$ . Observe that Line 1 of Algorithm 3 ensures that all edges of the bipartite input graph are cut.

To compute the approximation ratio of Algorithm 3, we also require the function

$$\delta_{\Theta}(x) := \frac{2 - \Theta(1 - x^+) + 2\sqrt{\Theta(x^+)(1 - \Theta(1 - x^+))}}{2 + 2x}. \quad (4.59)$$

For future reference, observe that

$$\delta_{\Theta}(x) = \frac{2 - \Theta(1)}{2 + 2x} \geq \frac{2 - \Theta(1)}{2} = \delta_{\Theta}(0) \quad \forall x \in (-1, 0], \quad (4.60)$$

which follows from the fact that  $x^+ = \max\{x, 0\} = 0$  for all  $x \in (-1, 0]$ , and  $\Theta(1) \leq 1$ .

**Lemma 4.29.** *Let  $G$  be a bipartite graph used as input to Algorithm 3,  $|\phi\rangle$  be as in (4.57),  $\Theta \in \mathcal{A}$  and  $h_{\max} \in [\sqrt{3}/2, 1]$ . Consider the values  $\theta_e$  as in (4.56). Suppose  $\theta_e = \arcsin \sqrt{\Theta(h_e^+)}$  for some edge  $e \in E(G)$  (i.e.,  $h_e \leq h_{\max}$ ), and  $h_e > -1$ . If for all edges  $e'$  adjacent to  $e$ ,  $\theta_{e'} = \arcsin \sqrt{\Theta(h_{e'}^+)}$ , then we have  $\frac{\langle \phi | H_e | \phi \rangle}{2 + 2h_e} \geq \delta_{\Theta}(h_e)$ .*

*Proof.* We substitute  $\theta_e = \arcsin \sqrt{\Theta(h_e^+)}$  in (4.58) (with  $e = \{i, j\}$ ), use that  $\sin \theta_e = \sqrt{\Theta(h_e^+)}$ , and apply (4.34), to obtain  $\langle \phi | H_e | \phi \rangle \geq 1 + (1 - \Theta(1 - h_e^+)) + 2\sqrt{\Theta(h_e^+)}\sqrt{1 - \Theta(1 - h_e^+)} = \delta_{\Theta}(h_e)(2 + 2h_e)$ . Since  $h_e > -1$ , the result follows from rewriting  $\langle \phi | H_e | \phi \rangle \geq \delta_{\Theta}(h_e)(2 + 2h_e)$ .  $\square$

Lastly, we require the following refinement of monogamy of entanglement, see (4.17).

**Lemma 4.30.** *Let  $e$  and  $e'$  be adjacent edges, and let  $L \in F_n^k$ , see (4.7), with  $k \geq 2$  and  $n \geq 3$ . Consider the  $h$  values as in Definition 4.8, corresponding to  $L$ . We have*

$$h_{e'} \leq \frac{1}{2} \left( \sqrt{3(1 - h_e^2)} - h_e \right). \quad (4.61)$$

*In particular,  $h_e \geq \frac{\sqrt{3}}{2} \implies h_{e'} \leq 0$ .*

*Proof.* [191, Lem. 3] shows that  $3(h_{e'} + h_e)^2 + (h_{e'} - h_e)^2 \leq 3$ , whenever  $h_{e'} + h_e \geq -1/2$  (recall that the variables in [191] are scaled differently compared to this chapter). Solving this inequality for  $h_{e'}$  yields (4.61). The implication  $h_e \geq \frac{\sqrt{3}}{2} \implies h_{e'} \leq 0$  follows directly from (4.61).  $\square$

We now compute (a lower bound on) the approximation ratio of Algorithm 3.

**Theorem 4.31.** *The approximation ratio of Algorithm 3 for bipartite graphs, with parameters  $\Theta \in \mathcal{A}$ ,  $h_{\max} \in [\sqrt{3}/2, 1]$  and  $\theta^* \in [0, 1]$ , is at least*

$$\min \left\{ \frac{2 - \theta^*}{2 + \sqrt{3}(1 - h_{\max}^2) - h_{\max}}, \min_{x \in [0, h_{\max}]} \delta_{\Theta}(x), \frac{1 + \sqrt{\theta^*}}{2} \right\}. \quad (4.62)$$

*Proof.* Let  $E$  be the edge set of the bipartite input graph. Let  $(h_e)_{e \in E}$  be the values obtained in Line 2 of Algorithm 3, and  $|\phi\rangle$  be as in (4.57). By the definition of  $H_G$  we have

$$\begin{aligned} \langle \phi | H_G | \phi \rangle &= \sum_{e \in E} w_e \langle \phi | H_e | \phi \rangle \geq \sum_{e \in E: h_e > -1} w_e \frac{\langle \phi | H_e | \phi \rangle}{2 + 2h_e} (2 + 2h_e) \\ &\geq \left( \inf_{e \in E: h_e > -1} \frac{\langle \phi | H_e | \phi \rangle}{2 + 2h_e} \right) \sum_{e \in E} w_e (2 + 2h_e) \geq \left( \inf_{e \in E: h_e > -1} \frac{\langle \phi | H_e | \phi \rangle}{2 + 2h_e} \right) \lambda_{\max}(H_G). \end{aligned} \quad (4.63)$$

For the first inequality, we have used that  $w_e \langle \phi | H_e | \phi \rangle \geq 0$ , since  $H_e = (1/4)H_e^2 \succeq 0$  and  $w_e > 0$ . The second inequality follows from the fact that  $\sum_{e \in E: h_e > -1} w_e (2 + 2h_e) = \sum_{e \in E} w_e (2 + 2h_e)$ . The last inequality follows from the fact that  $\text{SDP}^k$  provides an upper bound on  $\lambda_{\max}(H_G)$ , as explained in Section 4.2 (see also (4.14)).

Considering (4.63), the approximation ratio of Algorithm 3 is at least

$$\inf_{e \in E: h_e > -1} \frac{\langle \phi | H_e | \phi \rangle}{2 + 2h_e}. \quad (4.64)$$

Note here that  $\langle \phi | H_e | \phi \rangle$  is a function of  $h_e$ , which follows from the definition of  $|\phi\rangle$ , see (4.57). We show later that  $\lim_{h_e \downarrow -1} \frac{\langle \phi | H_e | \phi \rangle}{2 + 2h_e} = +\infty$ , which implies that (4.64) is well-defined. Let us determine a lower bound on (4.64) by distinguishing three cases, based on the value of  $h_e \in (-1, 1]$ .

**Case 1.**  $h_e < \frac{1}{2} \left( \sqrt{3(1 - h_{\max}^2)} - h_{\max} \right)$ .

Note that  $h_{\max} \geq \sqrt{3}/2$ , which implies that  $h_e < \frac{1}{2} \left( \sqrt{3(1 - h_{\max}^2)} - h_{\max} \right) \leq 0$ . Now  $h_e \leq 0 \implies h_e^+ = 0 \implies \theta_e = 0 \implies \sin \theta_e = 0$ . Consequently, the bound (4.58), with  $e = \{i, j\}$ , simplifies to

$$\langle \phi | H_e | \phi \rangle \geq 1 + \prod_{k \in N(i) \setminus \{j\}} \cos \theta_{ik} \prod_{k \in N(j) \setminus \{i\}} \cos \theta_{kj}. \quad (4.65)$$

Let  $E' := \{\{i, k\} : k \in N(i) \setminus \{j\}\}$ . By (4.17), at most one  $e' \in E'$  can satisfy  $h_{e'} > h_{\max} \geq \sqrt{3}/2$ . If there is one such  $e'$ , then by (4.61), all  $\tilde{e} \in E' \setminus \{e'\}$  satisfy

$h_{\bar{e}} \leq 0 \implies \theta_{\bar{e}} = 0 \implies \cos \theta_{\bar{e}} = 1$ . Note also that  $h_{e'} > h_{\max}$  implies that  $\theta_{e'} = \arcsin \sqrt{\theta^*}$ , see (4.56). Hence,  $\prod_{k \in N(i) \setminus \{j\}} \cos \theta_{ik} = \cos \theta_{e'} = \cos \arcsin \sqrt{\theta^*} = \sqrt{1 - \theta^*}$ .

Alternatively, if all  $e' \in E'$  satisfy  $h_{e'} \leq h_{\max}$ , we have that all  $\theta_{e'} = \arcsin \sqrt{\Theta(h_{e'}^+)}$ . Thus, we can proceed as in (4.34), and derive

$$\prod_{k \in N(i) \setminus \{j\}} \cos \theta_{ik} \geq \sqrt{1 - \Theta(1 - h_e^+)} = \sqrt{1 - \Theta(1)}.$$

To summarize both cases:

$$\prod_{k \in N(i) \setminus \{j\}} \cos \theta_{ik} \geq \min \left\{ \sqrt{1 - \theta^*}, \sqrt{1 - \Theta(1)} \right\} = \sqrt{1 - \max\{\theta^*, \Theta(1)\}}, \quad (4.66)$$

and similarly,  $\prod_{k \in N(j) \setminus \{i\}} \cos \theta_{kj} \geq \sqrt{1 - \max\{\theta^*, \Theta(1)\}}$ . We combine (4.65) and (4.66) to obtain  $\langle \phi | H_e | \phi \rangle \geq 2 - \max\{\theta^*, \Theta(1)\}$ . Consequently, (4.64) can be lower bounded by

$$\frac{\langle \phi | H_e | \phi \rangle}{2 + 2h_e} \geq \frac{2 - \max\{\theta^*, \Theta(1)\}}{2 + 2h_e} > \frac{2 - \max\{\theta^*, \Theta(1)\}}{2 + \sqrt{3(1 - h_{\max}^2)} - h_{\max}}. \quad (4.67)$$

Here, the last inequality is due to the assumption  $h_e < \frac{1}{2} \left( \sqrt{3(1 - h_{\max}^2)} - h_{\max} \right)$ , given by case 1. It follows by (4.67) that (4.64) is well-defined. Observe that

$$\begin{aligned} \sqrt{3(1 - h_{\max}^2)} - h_{\max} &\leq 0 \\ \implies \frac{2 - \Theta(1)}{2 + \sqrt{3(1 - h_{\max}^2)} - h_{\max}} &\geq \frac{2 - \Theta(1)}{2} = \delta_{\Theta}(0). \end{aligned} \quad (4.68)$$

For the second inequality in (4.68), we have used that  $\Theta(1) \leq 1$ , see (4.24). The equality in (4.68) follows from the definition of  $\delta_{\Theta}$ , see (4.59). By combining (4.67) and (4.68), it follows that

$$\frac{\langle \phi | H_e | \phi \rangle}{2 + 2h_e} \geq \min \left\{ \delta_{\Theta}(0), \frac{2 - \theta^*}{2 + \sqrt{3(1 - h_{\max}^2)} - h_{\max}} \right\}.$$

**Case 2.**  $h_e \in \left[ \frac{1}{2} \left( \sqrt{3(1 - h_{\max}^2)} - h_{\max} \right), h_{\max} \right]$ .

Let  $e'$  be an edge adjacent to  $e$ , and consider  $h_{e'}$ . If  $h_{e'} > h_{\max}$ , then  $h_e < \frac{1}{2} \left( \sqrt{3(1 - h_{\max}^2)} - h_{\max} \right)$ , see (4.61), which contradicts case 2. Thus, it holds that for all edges  $e'$  adjacent to  $e$ ,  $h_{e'} \leq h_{\max}$ . Now, by (4.56),  $h_{e'} \leq h_{\max} \implies \theta_{e'} = \arcsin \sqrt{\Theta(h_{e'}^+)}$ . Additionally,  $\theta_e = \arcsin \sqrt{\Theta(h_e^+)}$ . Hence, the conditions of Lemma 4.29 are satisfied, which implies that

$$\frac{\langle \phi | H_e | \phi \rangle}{2 + 2h_e} \geq \delta_{\Theta}(h_e) \geq \min_{x \in \left[ \frac{1}{2} \left( \sqrt{3(1 - h_{\max}^2)} - h_{\max} \right), h_{\max} \right]} \delta_{\Theta}(x) = \min_{x \in [0, h_{\max}]} \delta_{\Theta}(x).$$

Here, the equality is due to (4.60), and the fact that  $\frac{1}{2} \left( \sqrt{3(1-h_{\max}^2)} - h_{\max} \right) \leq 0$ , since  $h_{\max} \geq \sqrt{3}/2$ .

**Case 3.**  $h_e \in (h_{\max}, 1]$ .

Note that  $h_e > h_{\max} \geq \sqrt{3}/2$ . Hence, it follows from (4.61) that for all edges  $e'$  adjacent to  $e$ ,  $h_{e'} \leq 0 \implies h_{e'}^+ = 0 \implies \theta_{e'} = 0 \implies \cos \theta_{e'} = 1$ . Therefore, (4.58) simplifies to  $\langle \phi | H_e | \phi \rangle \geq 2 + 2 \sin \theta_e$ . Since  $h_e > h_{\max}$ , it follows by (4.56) that  $\theta_e = \arcsin \sqrt{\theta^*}$ , which implies that  $\sin \theta_e = \sqrt{\theta^*}$ . Consequently, (4.64) can be lower bounded by  $\frac{\langle \phi | H_e | \phi \rangle}{2+2h_e} \geq \frac{2+2 \sin \theta_e}{2+2h_e} = \frac{2+2\sqrt{\theta^*}}{2+2h_e} \geq \frac{2+2\sqrt{\theta^*}}{4} = \frac{1+\sqrt{\theta^*}}{2}$ , where the second inequality is due to  $h_e \leq 1$ .

By combining the three cases and noting that  $\delta_{\Theta}(0) \geq \min_{x \in [0, h_{\max}]} \delta_{\Theta}(x)$ , it follows that the value  $\frac{\langle \phi | H_e | \phi \rangle}{2+2h_e}$  is at least (4.62), which proves the theorem.  $\square$

The following result shows that the optimization problem  $\min_{x \in [0, h_{\max}]} \delta_{\Theta}(x)$  in (4.62), for  $\delta_{\Theta}$  as in (4.59), simplifies if  $\Theta(x) = Rx$  for some  $R \in [0, 1/2]$ . Recall from Lemma 4.26 that  $Rx \in \mathcal{A}$ .

**Lemma 4.32.** *Let  $h_{\max} \in [\sqrt{3}/2, 1]$  and  $\Theta(x) = Rx$ , with  $R \in [0, 1/2]$ . We have that  $\min_{x \in [0, h_{\max}]} \delta_{\Theta}(x) = \min \{ \delta_{\Theta}(0), \delta_{\Theta}(h_{\max}) \}$ .*

*Proof.* In case  $R = 0$ ,  $\delta_{\Theta}(x) = 2/(2+2x) \implies \min_{x \in [0, h_{\max}]} \delta_{\Theta}(x) = \delta_{\Theta}(h_{\max})$ . Let  $R \in (0, 1/2]$ . The derivative of  $\delta_{\Theta}(x)$  on the interval  $(0, 1]$ , is given by

$$\delta'_{\Theta}(x) = \frac{f_R(x)}{2(x+1)^2 \sqrt{Rx(Rx-R+1)}}, \quad (4.69)$$

where  $f_R(x) := (3R^2 - R)x + R - R^2 + (2R - 2)\sqrt{Rx(Rx - R + 1)}$ . A stationary point  $x^* \in (0, 1]$  of  $\delta_{\Theta}$  satisfies  $f_R(x^*) = 0$ . This is equivalent to  $((3R^2 - R)x^* + R - R^2)^2 - (2R - 2)^2 Rx^*(Rx^* - R + 1) = 0$ . The solution of this quadratic equation, on the interval  $(0, 1]$ , is given by  $x^* = \frac{R^3 + 2R^2 - 5R + 2 - 2(1-R)^2 \sqrt{1-R-R^2}}{R(R+1)(5R-3)}$ . It can be verified that for any  $R \in (0, 1/2]$ ,  $x^*$  is well-defined. Since the stationary point  $x^*$  is unique and  $\delta_{\Theta}$  is continuous, we have that

$$\min_{x \in [0, h_{\max}]} \delta_{\Theta}(x) = \min \{ \delta_{\Theta}(0), \delta_{\Theta}(x^*), \delta_{\Theta}(h_{\max}) \}. \quad (4.70)$$

By inspecting (4.69), it can be seen that there exists a positive  $\varepsilon$ , dependent on  $R$ , such that  $\delta'_{\Theta}(x) > 0$  for all  $x \in (0, \varepsilon]$ , which implies that  $\delta_{\Theta}(0) < \delta_{\Theta}(x^*)$ . By combining (4.70) and  $\delta_{\Theta}(0) < \delta_{\Theta}(x^*)$ , the result follows.  $\square$

We now provide input parameters for Algorithm 3, such that its approximation ratio is at least 0.8162.

**Theorem 4.33.** *Algorithm 3, for bipartite graphs, with inputs  $\Theta(x) = 0.367x$ ,  $h_{\max} = 0.876$  and  $\theta^* = 2/5$ , achieves an approximation ratio of at least 0.8162.*

*Proof.* The given parameters satisfy the requirements of Algorithm 3. By combining Theorem 4.31 and Lemma 4.32, we find that the approximation ratio of Algorithm 3 (with the given inputs) is at least

$$\min \left\{ \frac{2 - \theta^*}{2 + \sqrt{3}(1 - h_{\max}^2) - h_{\max}}, \delta_{\Theta}(0), \delta_{\Theta}(h_{\max}), \frac{1 + \sqrt{\theta^*}}{2} \right\}.$$

Using a computer, it can be verified that this value is at least 0.8162.  $\square$

Let

$$r^* := \max_{\substack{\Theta \in \mathcal{A}, \\ h_{\max} \in [\sqrt{3}/2, 1], \\ \theta^* \in [0, 1]}} \min \left\{ \frac{2 - \theta^*}{2 + \sqrt{3}(1 - h_{\max}^2) - h_{\max}}, \min_{x \in [0, h_{\max}]} \delta_{\Theta}(x), \frac{1 + \sqrt{\theta^*}}{2} \right\} \quad (4.71)$$

denote the optimal approximation ratio of Algorithm 3, with respect to the parameters  $\Theta$ ,  $h_{\max}$  and  $\theta^*$ . Theorem 4.33 proves that  $r^* \geq 0.8162$ . The following result provides an upper bound on  $r^*$  (similar to Theorem 4.27 for Algorithm 2).

**Theorem 4.34.** *The optimal approximation ratio of Algorithm 3, given by  $r^*$  as in (4.71), satisfies  $0.8162 \leq r^* < 0.8339$ .*

*Proof.* The lower bound on  $r^*$  is due to Theorem 4.33. For the upper bound, note first that for any  $\Theta \in \mathcal{A}$ ,  $h_{\max} \in [\sqrt{3}/2, 1]$  and  $\theta^* \in [0, 1]$ , we have

$$\begin{aligned} & \min \left\{ \frac{2 - \theta^*}{2 + \sqrt{3}(1 - h_{\max}^2) - h_{\max}}, \min_{x \in [0, h_{\max}]} \delta_{\Theta}(x), \frac{1 + \sqrt{\theta^*}}{2} \right\} \\ & \leq \min_{x \in [0, h_{\max}]} \delta_{\Theta}(x) \leq \min_{x \in [0, \sqrt{3}/2]} \delta_{\Theta}(x) \leq \min \left\{ \delta_{\Theta}(0), \delta_{\Theta}\left(\frac{2 - \sqrt{3}}{2}\right), \delta_{\Theta}\left(\frac{\sqrt{3}}{2}\right) \right\}. \end{aligned} \quad (4.72)$$

Here, the second inequality follows from  $h_{\max} \in [\sqrt{3}/2, 1]$ , which implies that  $[0, \sqrt{3}/2] \subseteq [0, h_{\max}]$ . The last inequality follows from the fact that  $\delta_{\Theta}(y) \geq \min_{x \in [0, \sqrt{3}/2]} \delta_{\Theta}(x)$  for any  $y \in [0, \sqrt{3}/2]$ . By combining (4.72) with the definition of  $r^*$ , we have

$$r^* \leq \max_{\Theta \in \mathcal{A}} \min \left\{ \delta_{\Theta}(0), \delta_{\Theta}\left(\frac{2 - \sqrt{3}}{2}\right), \delta_{\Theta}\left(\frac{\sqrt{3}}{2}\right) \right\}. \quad (4.73)$$

Eq. (4.73) is fully determined by  $z := (z_1, z_2, z_3) = \left( \Theta\left(\frac{2 - \sqrt{3}}{2}\right), \Theta\left(\frac{\sqrt{3}}{2}\right), \Theta(1) \right)$ , i.e.,  $\delta_{\Theta}(0) = 1 - \frac{z_3}{2}$ ,  $\delta_{\Theta}\left(\frac{2 - \sqrt{3}}{2}\right) = \frac{2 - z_2 + 2\sqrt{z_1(1 - z_2)}}{4 - \sqrt{3}}$ ,  $\delta_{\Theta}\left(\frac{\sqrt{3}}{2}\right) = \frac{2 - z_1 + 2\sqrt{z_2(1 - z_1)}}{2 + \sqrt{3}}$ , which follows from the definition of  $\delta_{\Theta}$ , see (4.59). By the properties of functions in  $\mathcal{A}$ , see (4.24), we have that  $1 - z_3 = 1 - \Theta(1) = \min_{x \in [0, 1]} (1 - \Theta(x))(1 - \Theta(1 - x)) \leq$

$\left(1 - \Theta\left(\frac{2-\sqrt{3}}{2}\right)\right) \left(1 - \Theta\left(\frac{\sqrt{3}}{2}\right)\right) = (1 - z_1)(1 - z_2)$ . Additionally, since  $\Theta$  is an increasing function,  $0 = \Theta(0) \leq z_1 \leq z_2 \leq z_3 = \Theta(1) \leq 1$ . We define

$$F := \{z \in \mathbb{R}^3 : 1 - z_3 \leq (1 - z_1)(1 - z_2), 0 \leq z_1 \leq z_2 \leq z_3 \leq 1\} \quad (4.74)$$

as the set of  $z$  that satisfy the previously derived constraints. Using (4.73) and  $F$ , we derive the following upper bound on  $r^*$ , in terms of  $z$ :

$$r^* \leq \max_{z \in F} \min \left\{ 1 - \frac{z_3}{2}, \frac{2 - z_2 + 2\sqrt{z_1(1 - z_2)}}{4 - \sqrt{3}}, \frac{2 - z_1 + 2\sqrt{z_2(1 - z_1)}}{2 + \sqrt{3}} \right\}. \quad (4.75)$$

We define  $r := 0.8339$ . It follows by (4.75) that  $r^* < r$  is implied by the inconsistency of the following system of equations:

$$\begin{aligned} z \in F, \quad 1 - \frac{z_3}{2} \geq r, \quad \frac{2 - z_2 + 2\sqrt{z_1(1 - z_2)}}{4 - \sqrt{3}} \geq r, \\ \frac{2 - z_1 + 2\sqrt{z_2(1 - z_1)}}{2 + \sqrt{3}} \geq r. \end{aligned} \quad (4.76)$$

We will show that (4.76) is inconsistent. Observe that  $1 - z_3/2 \geq r \implies z_3 \leq 2(1 - r)$ . We may assume without loss of generality that any solution to (4.76), if it exists, satisfies  $z_3 = 2(1 - r)$ . Indeed, if  $(z_1, z_2, z_3)$  is a solution to (4.76), also  $(z_1, z_2, 2(1 - r))$  is a solution to (4.76). Thus, the inconsistency of (4.76) is equivalent to the inconsistency of the following system of equations:

$$\begin{aligned} 0 \leq z_1 \leq z_2 \leq 2(1 - r), \quad (1 - z_1)(1 - z_2) \geq 2r - 1, \\ 2\sqrt{z_1(1 - z_2)} - z_2 \geq (4 - \sqrt{3})r - 2, \\ 2\sqrt{z_2(1 - z_1)} - z_1 \geq (2 + \sqrt{3})r - 2, \end{aligned} \quad (4.77)$$

where the first line of (4.77) ensures that  $z \in F$ , see (4.74). Lemma B.10 on Page 215 proves that (4.77) is inconsistent. Hence, also (4.76) is inconsistent, which implies that  $r^* < r$ , proving the result.  $\square$

## 4.6 Conclusions

In this chapter, we study classical approximation algorithms for the QMC problem on general, triangle-free, and bipartite graphs. For triangle-free and bipartite graphs, we introduce new approximation algorithms. We prove that the algorithms achieve approximation ratios of at least 0.603 (general graphs) 0.61383 (triangle-free graphs) and 0.8162 (bipartite graphs) respectively. For the QMC problem on triangle-free graphs, and on bipartite graphs, these ratios are the current best for their respective problems.

The key part of the analysis of the algorithm for general graphs is showing that a particular vector induced by the used QMC SDP relaxation is contained in the

matching polytope. We show this by introducing a graph parameter  $c(G, k)$ , see (4.15), for SDP relaxation level  $k \in \mathbb{N}$ , and verifying that  $c(G, k) \leq \lfloor s/2 \rfloor$  for all graphs  $G$  on  $s$  vertices. We establish properties of  $c(G, k)$  that greatly reduce the required computation time of verifying this inequality (see Lemmas 4.15 and 4.16), and prove (using a computer) that  $c(G, s) \leq \lfloor s/2 \rfloor$  holds for all graphs on  $s$  vertices, where  $s$  is odd and  $s \leq 13$ . As future work, it would be interesting to determine if this extends to odd values of  $s > 13$ , which, if so, results in an improved approximation ratio. A possible starting point in this direction is to consider the 5-cycle, which is the smallest graph for which we have no analytical proof of  $c(G, 2) \leq \lfloor s/2 \rfloor$  (see Table 4.3).

The studied QMC approximation algorithms for triangle-free and bipartite graphs both require a function  $\Theta \in \mathcal{A}$ , see (4.24), as parameter. For the triangle-free algorithm, we provide a function  $\Theta$  that achieves an approximation ratio that is 0.00009 below the optimal ratio (see Theorem 4.27). For the bipartite algorithm, the larger gap of 0.0177 (see Theorem 4.34) motivates further study. For both algorithms, an optimal  $\Theta \in \mathcal{A}$  does not achieve an approximation ratio of 0.956, which is the optimal QMC approximation ratio under UGC and a related conjecture, see [146]. It is therefore interesting to investigate if the constraint  $\Theta \in \mathcal{A}$  can be relaxed.

## 5 SDP bounds on the stability number via ADMM and intermediate levels of the Lasserre hierarchy

A stable set in a graph is a subset of vertices that are pairwise non-adjacent. Given an undirected graph  $G$ , the stable set problem is to determine a stable set in  $G$  of maximum cardinality. The stability number of  $G$ , denoted by  $\alpha(G)$ , is defined as the cardinality of a maximum stable set in  $G$ . Computing  $\alpha(G)$  is NP-hard. Hence, unless  $P = NP$ , there is no polynomial time algorithm that computes it.

There exist different approaches for computing upper bounds on the stability number of a graph, and one of those is using semidefinite programming. The first SDP relaxation of the stable set problem is due to Lovász [204], who introduced the Lovász theta function of a graph  $G$ , denoted by  $\vartheta(G)$ . For any graph  $G$ ,  $\vartheta(G) \geq \alpha(G)$  and  $\vartheta(G)$  can be computed in polynomial time up to fixed precision. Semidefinite programs (SDPs) that define the Lovász theta function can be strengthened by cutting planes, see e.g., [78, 128, 260, 271]. The paper by Pucher and Rendl [260] currently provides one of the strongest SDP-based bounds for the stable set problem.

Several hierarchical approaches can also be applied to construct relaxations of the stable set problem. Higher levels in the hierarchy correspond to stronger relaxations, which are also more difficult to solve due to the increased number of variables and constraints. Among the hierarchies that can be applied to the stable set problem are the Sherali-Adams hierarchy [274] (based on linear programming), the SDP-based hierarchy of Lovász-Schrijver [205], the Lasserre hierarchy [175], and the exact subgraph hierarchy (ESH) [3]. Much research has recently been devoted to the ESH for the stable set problem [97, 98, 99]. The numerical results in those papers show that the ESH provides strong bounds within a reasonable computational time. It is known that the Lasserre hierarchy is stronger than the other hierarchies [182, 277]. Despite this, little research has been devoted to the practical performance of the Lasserre hierarchy for the stable set problem. A practical drawback of the Lasserre hierarchy is the order of the associated positive semidefinite (PSD) matrix variable: level  $k$  of the hierarchy involves a PSD matrix variable of order  $\mathcal{O}(n^k)$ , where  $n$  is the number of vertices in the graph.

The classical method for solving semidefinite programs (SDPs) is the interior-point method (IPM) [7, 236]. IPMs typically require large amounts of memory, which limits their applicability to the Lasserre hierarchy. The IPM is a second-order method that

requires the construction and Cholesky factorization of an  $m \times m$  Schur complement matrix, where  $m$  is the number of linear equality constraints in the SDP relaxation. Constructing and storing this dense matrix requires substantial memory, particularly when  $m$  is large, such as in SDPs derived from the Lasserre hierarchy. The worst-case complexity of computing the Schur complement matrix is  $\mathcal{O}(mp^3 + m^2p^2)$  [227], where  $p$  is the order of the PSD variable. Computational complexity may be reduced in cases where the constraint matrices exhibit special structures such as low rank, see e.g., [65, 148]. The Cholesky factorization for the Schur complement matrix requires  $\mathcal{O}(m^3)$  operations, which becomes impractical for large values of  $m$ . These limitations of IPMs motivate the use of alternative methods that require less memory and bypass the Cholesky factorization.

The alternating direction method of multipliers (ADMM), see e.g., [34], is a first-order method that can be used to solve SDPs, and requires significantly less memory than IPMs. Each iteration of the ADMM algorithm for solving SDPs consists of three steps: the orthogonal projection onto the cone of positive semidefinite matrices, an orthogonal projection onto a polyhedral set, and a dual update step. The most memory intensive step of the ADMM is computing the eigendecomposition of a symmetric PSD matrix of order  $p$ , in each main loop of the algorithm. The SDP relaxations of the stable set problem arising from the Lasserre hierarchy satisfy  $m \gg p$ , and the computational complexity of eigendecomposition for a symmetric matrix of order  $p$  is  $\mathcal{O}(p^3)$ . Considering this, along with the fact that projections onto the polyhedral sets from the Lasserre relaxations can be performed efficiently, ADMMs appear more suitable than IPMs for computing Lasserre bounds for the stable set problem.

In this chapter, we bridge the gap between theory and practice by using the ADMM to effectively compute Lasserre hierarchy bounds for the stability number. In particular, we compute bounds from intermediate levels of the Lasserre hierarchy for  $k$  between 1 and 2, including  $k = 2$ , on graphs with at most 300 vertices. However, we are not the first to consider intermediate levels of the Lasserre hierarchy; these have been employed in several papers, see e.g., [10, 49, 54, 282, 299]. For the majority of graphs, we restrict ourselves to intermediate levels of the hierarchy due to practical limitations on the order of the considered PSD variable, which we cap at 2500. Eigendecomposition for a symmetric matrix of that order can still be performed reasonably well, especially when single precision is used.

The number of inequality constraints in the resulting SDP relaxation depends on the considered graph and is at most 2,510,148 in our experiments. Storing the  $m \times m$  Schur complement matrix (in standard double precision), with  $m = 2,510,148$ , requires approximately 47,000 GB of RAM (!). Therefore, IPMs are intractable for solving the corresponding relaxations.

Constructing an intermediate-level SDP relaxation of the Lasserre hierarchy for  $k$  between 1 and 2 requires selecting specific degree two monomials. These monomials of degree two, along with all monomials of degree one and the monomial of degree zero, form a basis used to derive the SDP relaxation. We present a basis selection method that exploits an SDP relaxation of the Lovász theta function. It is known that  $\vartheta(G)$  corresponds to the first level of the Lasserre hierarchy applied to the stable set problem, see e.g., [182, Sect. 6]. Our ADMM algorithm incorporates a warm-starting

approach to further improve performance. The numerical results show that the upper bounds on  $\alpha(G)$  computed here are competitive with the best SDP-based bounds for the stable set problem. Moreover, these bounds can be obtained using the ADMM within reasonable running times, specifically within one hour.

This chapter is organized as follows. Preliminaries are provided in Section 5.1. We provide details of the Lasserre hierarchy for the stable set problem in Section 5.2. In Section 5.3, we show how to apply the ADMM to the SDPs arising from the Lasserre hierarchy. Section 5.4 presents a basis selection method for the construction of intermediate levels of the Lasserre hierarchy, and provides a method for warm-starting the ADMM. Numerical results are presented in Section 5.5, and conclusions are provided in Section 5.6.

## 5.1 Preliminaries

Given  $n, k \in \mathbb{N}$ , we define  $[n] := \{1, \dots, n\}$  and  $\binom{[n]}{\leq k} := \{\beta \subseteq [n] : |\beta| \leq k\}$ . Let  $\mathcal{B} \subseteq \binom{[n]}{\leq k}$ . For any such  $\mathcal{B}$ , we define

$$\mathcal{B}^{2\cup} := \{\beta \cup \beta' : \beta, \beta' \in \mathcal{B}\}. \quad (5.1)$$

For any  $\beta \subseteq [n]$ , we define  $\mathbb{1}_\beta \in \{0, 1\}^n$  as the indicator vector corresponding to  $\beta$ , i.e.,  $(\mathbb{1}_\beta)_i = 1$  if and only if  $i \in \beta$ . We define  $\mathbb{I}(\beta) \in \{0, 1\}$  as the indicator that equals 1 if the cardinality  $|\beta| = 1$ , and 0 otherwise.

Given  $\mathcal{B} \subseteq \binom{[n]}{\leq k}$ , we define  $\mathbf{x}^\mathcal{B}$  as the  $|\mathcal{B}|$ -dimensional vector of all monomials  $x^\beta := \prod_{i \in \beta} x_i$ ,  $\beta \in \mathcal{B}$ , with  $x^\emptyset = 1$ . The monomials of the vector  $\mathbf{x}^\mathcal{B} = (x^\beta)_{\beta \in \mathcal{B}}$  form a basis of some subspace of  $\mathbb{R}[x]$ . With slight abuse of terminology, we refer to both  $\mathbf{x}^\mathcal{B}$  and  $\mathcal{B}$  as a bases.

## 5.2 The Lasserre hierarchy for the stable set problem

We present the Lasserre hierarchy [175] for the stable set problem. Similar derivations can also be found in, e.g., [120, Sect. 3.1], [184], and [185, Example 8.16].

Let  $G = (V, E)$  be a simple undirected graph. Without loss of generality, we assume that  $V = [n]$  for some  $n \in \mathbb{N}$ . Let  $\beta \subseteq [n]$ . If  $\beta$  is a stable set in  $G$ , we say that  $\beta$  is stable in  $G$ . We define  $[n]_G := \{\beta \subseteq [n] : \beta \text{ is stable in } G\}$  as the set of all stable sets in  $G$ , and set  $S_G := \{\mathbb{1}_\beta : \beta \in [n]_G\}$ . We define  $\mathbb{P}_G \subseteq \mathbb{R}[x]$  as the set of polynomials nonnegative over  $S_G$ .

Observe that the stability number  $\alpha(G) = \max_{x \in S_G} \sum_{i \in [n]} x_i$ , which is equivalent to

$$\alpha(G) = \min_{\mu \in \mathbb{R}} \left\{ \mu : \mu - \sum_{i \in [n]} x_i \in \mathbb{P}_G \right\}. \quad (5.2)$$

This equivalence follows from the observation that, for fixed  $\mu$ , we have  $\min_{x \in S_G} \mu - \sum_{i \in [n]} x_i = \mu - \max_{x \in S_G} \sum_{i \in [n]} x_i = \mu - \alpha(G)$ , which implies that  $\mu - \sum_{i \in [n]} x_i$  is a

nonnegative polynomial over  $S_G$  if and only if  $\mu \geq \alpha(G)$ . In general, it is NP-hard to optimize over  $S_G$  or  $\mathbb{P}_G$ , which motivates the search of tractable relaxations of (5.2). Let us formulate such a relaxation, in terms of sum of squares (SOS) polynomials. To this end, we define the polynomial ideal

$$\mathcal{I}_G := \langle x_i^2 - x_i \text{ for all } i \in [n], x_i x_j \text{ for all } \{i, j\} \in E \rangle, \quad (5.3)$$

that is used to define  $\mathbb{P}_G$  as follows:

$$\mathbb{P}_G := \left\{ f \in \mathbb{R}[x] : f \equiv \sum_{j \in [k]} f_j^2 \pmod{\mathcal{I}_G}, f_j \in \mathbb{R}[x] \text{ for all } j \in [k], k \in \mathbb{N} \right\},$$

see e.g., [248, Thm. 1]. Note that  $\mathcal{I}_G$  encodes the set  $S_G$  in the sense that  $x \in S_G$  if and only if  $f(x) = 0$  for all  $f \in \mathcal{I}_G$ . Polynomials of the form  $\sum_{j \in [k]} f_j^2$  are called SOS polynomials. SOS polynomials, and thus polynomials in  $\mathbb{P}_G$ , can be expressed in terms of PSD matrices. Specifically, we have that

$$f \in \mathbb{P}_G \iff f \equiv (\mathbf{x}^{\mathcal{B}'})^\top A \mathbf{x}^{\mathcal{B}'} \pmod{\mathcal{I}_G} \text{ for some } A \in \mathcal{S}_+^{|\mathcal{B}'|},$$

where  $\mathcal{B}' := \binom{[n]}{\leq n}$ , see e.g., [178, Prop. 2.1]. This shows that one can optimize over  $\mathbb{P}_G$  using semidefinite programming. However,  $|\mathcal{B}'|$  is exponential in  $n$ , which makes  $\mathbb{P}_G$  intractable. It is therefore natural to consider a subset of  $\mathbb{P}_G$  by fixing a  $\mathcal{B} \subseteq \binom{[n]}{\leq n}$  such that  $|\mathcal{B}|$  is polynomial in  $n$ . For any  $\binom{[n]}{\leq 1} \subseteq \mathcal{B} \subseteq \binom{[n]}{\leq n}$ , we define a corresponding subset of  $\mathbb{P}_G$  as

$$\mathbb{P}_G(\mathcal{B}) := \left\{ f \in \mathbb{R}[x] : \begin{array}{l} f \equiv (\mathbf{x}^{\mathcal{B}})^\top A \mathbf{x}^{\mathcal{B}} + c^\top \mathbf{x}^{\mathcal{B}^{2\cup}} \pmod{\mathcal{I}_G}, \\ A \in \mathcal{S}_+^{|\mathcal{B}|}, c \in \mathbb{R}_+^{|\mathcal{B}^{2\cup}|} \end{array} \right\}, \quad (5.4)$$

for  $\mathcal{B}^{2\cup}$  as in (5.1). In (5.4), the term  $c^\top \mathbf{x}^{\mathcal{B}^{2\cup}}$  is introduced to enlarge  $\mathbb{P}_G(\mathcal{B})$ , resulting in stronger bounds on  $\alpha(G)$ . Moreover, we show in Section 5.3.1 that the addition of this term does not increase the computational cost of obtaining these bounds. Note that  $c^\top \mathbf{x}^{\mathcal{B}^{2\cup}}$  is an SOS polynomial modulo  $\mathcal{I}_G$ , since  $c^\top \mathbf{x}^{\mathcal{B}^{2\cup}} \equiv (\mathbf{x}^{\mathcal{B}^{2\cup}})^\top \text{Diag}(c) \mathbf{x}^{\mathcal{B}^{2\cup}} \pmod{\mathcal{I}_G}$ , and  $\text{Diag}(c) \succeq 0$ . Since  $(\mathbf{x}^{\mathcal{B}})^\top A \mathbf{x}^{\mathcal{B}}$  is also an SOS polynomial, it follows that  $\mathbb{P}_G(\mathcal{B}) \subseteq \mathbb{P}_G$ . From (5.2), we observe that the value

$$\alpha^{\mathcal{B}}(G) := \min_{\mu \in \mathbb{R}} \left\{ \mu : \mu - \sum_{i \in [n]} x_i \in \mathbb{P}_G(\mathcal{B}) \right\} \quad (5.5)$$

satisfies  $\alpha^{\mathcal{B}}(G) \geq \alpha(G)$ , since  $\mathbb{P}_G(\mathcal{B}) \subseteq \mathbb{P}_G$ . Note that  $\alpha^{\mathcal{B}}(G)$  is well-defined since  $\binom{[n]}{\leq 1} \subseteq \mathcal{B}$ . Computing  $\alpha^{\mathcal{B}}(G)$  is equivalent to solving an SDP (see Section 5.3.1) wherein the PSD variable is of order  $|\mathcal{B}|$ . If  $|\mathcal{B}|$  is polynomial in  $n$ , the value  $\alpha^{\mathcal{B}}(G)$  can be computed in polynomial time up to fixed precision [262, Cor. 9].

It is worth noting that the computational effort of computing  $\alpha^{\mathcal{B}}(G)$  for some  $\mathcal{B} \subseteq \binom{[n]}{\leq n}$  can be reduced significantly by computing instead  $\alpha^{\mathcal{B} \cap [n]_G}(G)$ . Indeed,

$|\mathcal{B} \cap [n]_G| \leq |\mathcal{B}|$ , which results in a smaller PSD variable, and  $\alpha^{\mathcal{B}}(G) = \alpha^{\mathcal{B} \cap [n]_G}(G)$ . This equality follows from (the more general) [184, Cor. 16]. We present a proof here for the sake of completion.

**Lemma 5.1.** *Let  $G$  be a graph on  $n$  vertices and  $\mathcal{B} \subseteq \binom{[n]}{\leq n}$ . For  $\alpha^{\mathcal{B}}(G)$  as in (5.5), we have that  $\alpha^{\mathcal{B}}(G) = \alpha^{\mathcal{B} \cap [n]_G}(G)$ .*

*Proof.* For notational convenience, we write  $\mathcal{B}' := \mathcal{B} \cap [n]_G$ . By the definition of  $\alpha^{\mathcal{B}}(G)$ , it suffices to show that  $\mathbb{P}_G(\mathcal{B}') = \mathbb{P}_G(\mathcal{B})$ . Since  $\mathcal{B}' \subseteq \mathcal{B}$ , it is clear from (5.4) that  $\mathbb{P}_G(\mathcal{B}') \subseteq \mathbb{P}_G(\mathcal{B})$ . Thus, it remains to show the reverse inclusion. Let  $f \in \mathbb{P}_G(\mathcal{B})$ , and let  $A \in \mathcal{S}_+^{|\mathcal{B}|}$  and  $c \in \mathbb{R}_+^{|\mathcal{B}^{2\cup}|}$ , see (5.1), satisfy  $f \equiv (\mathbf{x}^{\mathcal{B}})^\top A \mathbf{x}^{\mathcal{B}} + c^\top \mathbf{x}^{\mathcal{B}^{2\cup}} \pmod{\mathcal{I}_G}$ , where  $\mathcal{I}_G$  is defined in (5.3). Let  $A$  be indexed by the elements of  $\mathcal{B}$ , and let  $A'$  be the principal submatrix of  $A$  indexed by the elements of  $\mathcal{B}'$ . We define similarly the vector  $c' = (c_\beta)_{\beta \in (\mathcal{B}')^{2\cup}}$ . Since  $x^\beta \equiv 0 \pmod{\mathcal{I}_G}$  for any  $\beta \in \mathcal{B} \setminus \mathcal{B}'$ , we have that

$$f \equiv (\mathbf{x}^{\mathcal{B}})^\top A \mathbf{x}^{\mathcal{B}} + c^\top \mathbf{x}^{\mathcal{B}^{2\cup}} \equiv (\mathbf{x}^{\mathcal{B}'})^\top A' \mathbf{x}^{\mathcal{B}'} + (c')^\top \mathbf{x}^{(\mathcal{B}')^{2\cup}} \pmod{\mathcal{I}_G}.$$

Hence,  $f \in \mathbb{P}_G(\mathcal{B}')$ , which completes the proof.  $\square$

The  $k$ th level of the Lasserre hierarchy for the stable set problem is to compute  $\alpha^{\binom{[n]}{\leq k}}(G)$ , or equivalently,  $\alpha^{\binom{[n]}{\leq k} \cap [n]_G}(G)$ . For any fixed value of  $k$ ,  $\left| \binom{[n]}{\leq k} \right| \in \mathcal{O}(n^k)$ , and thus,  $\alpha^{\binom{[n]}{\leq k}}(G)$  can be computed in polynomial time. The sequence  $(\alpha^{\binom{[n]}{\leq k}}(G))_{k \in \mathbb{N}}$  is decreasing towards  $\alpha(G)$ . Moreover, if  $\alpha(G) \geq 2$ , then  $\alpha^{\binom{[n]}{\leq k}}(G) = \alpha(G)$  for  $k \geq \alpha(G) - 1$  [182, Prop. 21].

## 5.3 The ADMM for computing $\alpha^{\mathcal{B}}(G)$

In this section we propose the ADMM for computing  $\alpha^{\mathcal{B}}(G)$ , for general bases  $\binom{[n]}{\leq 1} \subseteq \mathcal{B} \subseteq \binom{[n]}{\leq n}$ . The ADMM has been successfully used to solve SDPs, see e.g., [211, 243, 266, 282]. Compared to the IPM, the classical method for solving SDPs, the ADMM requires less memory, making it better suited for solving the large-scale SDPs that arise in the Lasserre hierarchy for the stable set problem.

Throughout the remainder of this section we fix some basis  $\mathcal{B}$  that satisfies  $\binom{[n]}{\leq 1} \subseteq \mathcal{B} \subseteq \binom{[n]}{\leq n}$ .

### 5.3.1 The SDP defining $\alpha^{\mathcal{B}}(G)$

Consider the constraint  $\mu - \sum_{i \in [n]} x_i \in \mathbb{P}_G(\mathcal{B})$  in the definition of  $\alpha^{\mathcal{B}}(G)$ , given by (5.5). It follows from (5.4) that this constraint is equivalent to

$$\mu - \sum_{i \in [n]} x_i \equiv (\mathbf{x}^{\mathcal{B}})^\top A \mathbf{x}^{\mathcal{B}} + c^\top \mathbf{x}^{\mathcal{B}^{2\cup}} \pmod{\mathcal{I}_G}, \quad A \in \mathcal{S}_+^{|\mathcal{B}|}, \quad c \in \mathbb{R}_+^{|\mathcal{B}^{2\cup}|}, \quad (5.6)$$

where  $\mathcal{B}^{2\cup}$  is defined in (5.1), and  $\mathcal{I}_G$  in (5.3). Let us index the matrix  $A$  with the elements of  $\mathcal{B}$ , and  $c$  with elements of  $\mathcal{B}^{2\cup}$ .

The equivalence relation (5.6) implies that the two polynomials have equal coefficients of  $x^\beta$ , for all  $\beta \in [n]_G$ . Thus, (5.6) implies the following linear equalities:  $A_{\emptyset,\emptyset} + c_\emptyset = \mu$  and

$$\sum_{\beta,\beta' \in \mathcal{B}:\beta \cup \beta' = \gamma} A_{\beta,\beta'} + c_\gamma = -\mathbb{I}(\gamma) \text{ for all } \gamma \in \mathcal{B}^{2\cup} \cap [n]_G, \gamma \neq \emptyset. \quad (5.7)$$

Here, we consider  $\gamma \in \mathcal{B}^{2\cup}$  since for any  $\beta, \beta' \subseteq [n]$ ,  $x^\beta x^{\beta'} \equiv x^{\beta \cup \beta'} \pmod{\mathcal{I}_G}$ . Since the objective in (5.5) is to minimize  $\mu$ , it follows that at optimality, we have  $A_{\emptyset,\emptyset} = \mu$  and  $c_\emptyset = 0$ . We eliminate the other entries of  $c$ , by transforming the equality constraints from (5.7) into inequality constraints, using that  $c \geq 0$ . It then follows that

$$\alpha^{\mathcal{B}}(G) = \min \left\{ A_{\emptyset,\emptyset} : A \in \mathcal{S}_+^{|\mathcal{B}|} \cap \mathcal{F}(\mathcal{B}) \right\}, \quad (5.8)$$

where

$$\mathcal{F}(\mathcal{B}) := \left\{ A \in \mathcal{S}^{|\mathcal{B}|} : \sum_{\substack{\beta,\beta' \in \mathcal{B}:\beta \cup \beta' = \gamma \\ \gamma \in \mathcal{B}^{2\cup} \cap [n]_G, \gamma \neq \emptyset}} A_{\beta,\beta'} \leq -\mathbb{I}(\gamma) \text{ for all } \right\}. \quad (5.9)$$

We will use formulation (5.8) to compute  $\alpha^{\mathcal{B}}(G)$  via the ADMM.

### 5.3.2 The ADMM iterates

To apply the ADMM to (5.8), we first reformulate (5.8) as the following optimization problem:

$$\begin{aligned} \min_{X, Y \in \mathcal{S}^{|\mathcal{B}|}} \quad & Y_{\emptyset,\emptyset} \\ \text{s.t.} \quad & X \in \mathcal{S}_+^{|\mathcal{B}|}, Y \in \mathcal{F}(\mathcal{B}), X = Y, \end{aligned} \quad (5.10)$$

where  $\mathcal{F}(\mathcal{B})$  is defined in (5.9). Given a penalty parameter  $\rho > 0$ , the augmented Lagrangian associated to (5.10), with respect to the constraint  $X = Y$ , is the function

$$\mathcal{L}_\rho(X, Y, Z) := Y_{\emptyset,\emptyset} + \rho \langle Z, Y - X \rangle + \frac{\rho}{2} \|Y - X\|^2, \quad (5.11)$$

where  $Z \in \mathcal{S}^{|\mathcal{B}|}$  is the (scaled) dual variable. Given some initial  $X^1, Y^1, Z^1 \in \mathcal{S}^{|\mathcal{B}|}$ , the ADMM computes the sequence of matrices  $(X^\ell, Y^\ell, Z^\ell)_{\ell \in \mathbb{N}}$ , defined recursively as

$$\begin{aligned} X^{\ell+1} &:= \arg \min_{X \succeq 0} \mathcal{L}_\rho(X, Y^\ell, Z^\ell) \\ Y^{\ell+1} &:= \arg \min_{Y \in \mathcal{F}(\mathcal{B})} \mathcal{L}_\rho(X^{\ell+1}, Y, Z^\ell) \\ Z^{\ell+1} &:= Z^\ell + \nu (Y^{\ell+1} - X^{\ell+1}), \end{aligned} \quad (5.12)$$

where  $\nu \in \mathbb{R}$  is a stepsize parameter. The matrices  $X^\ell$  and  $Y^\ell$  converge with rate  $\mathcal{O}(1/\ell)$  (in the ergodic sense) to an optimal solution of (5.10) [136, Thm. 6.5] when  $\nu \in \left(0, \frac{1+\sqrt{5}}{2}\right)$  [90, Thm. 5.1].

The minimization problems in (5.12) admit the following closed form solutions, see e.g., [243, Eq. 3.4]:

$$\begin{aligned} \arg \min_{X \succeq 0} \mathcal{L}_\rho(X, Y^\ell, Z^\ell) &= \mathcal{P}_{\mathcal{S}_+^{|\mathcal{B}|}}(Y^\ell + Z^\ell), \\ \arg \min_{Y \in \mathcal{F}(\mathcal{B})} \mathcal{L}_\rho(X^{\ell+1}, Y, Z^\ell) &= \mathcal{P}_{\mathcal{F}(\mathcal{B})}\left(X^{\ell+1} - \frac{1}{\rho}H - Z^\ell\right), \end{aligned} \quad (5.13)$$

where  $H \in \mathcal{S}^{|\mathcal{B}|}$  is the matrix that is zero everywhere, except for the entry  $H_{\emptyset, \emptyset} = 1$ . Note that  $\langle H, X \rangle = X_{\emptyset, \emptyset}$ . The augmented Lagrangian (5.11) and scheme (5.12) correspond to the *scaled form* of the ADMM, see e.g., [34, Sect. 3.1.1] or Appendix A.1 on Page 191. Compared to the unscaled form, the scaled form of the ADMM avoids the multiplication of  $(1/\rho)$  with  $Z^\ell$ , see (5.13).

We briefly discuss the two projections in (5.13). For any  $A \in \mathcal{S}^{|\mathcal{B}|}$ ,

$$\mathcal{P}_{\mathcal{S}_+^{|\mathcal{B}|}}(A) = \sum_{\lambda \in \Lambda(A): \lambda > 0} \lambda u_\lambda u_\lambda^\top, \quad (5.14)$$

where  $\Lambda(A)$  denotes the eigenspectrum of  $A$ , and the vectors  $(u_\lambda)_{\lambda \in \Lambda(A)}$  form an orthonormal basis of eigenvectors. To project a symmetric matrix  $Y$  onto  $\mathcal{F}(\mathcal{B})$ , see (5.9), we consider  $\mathcal{F}(\mathcal{B})$  as a set of vectors by identifying  $Y$  with its upper triangular entries. To account for the symmetry of  $Y$ , we replace the terms  $Y_{\beta, \beta'} + Y_{\beta', \beta}$  with  $2Y_{\beta, \beta'}$ . From this point of view, it can be seen that  $\mathcal{F}(\mathcal{B})$  is (up to reordering) a Cartesian product of closed half-spaces, one for each nonempty  $\gamma \in \mathcal{B}^{2\cup} \cap [n]_G$ . Therefore, projecting onto  $\mathcal{F}(\mathcal{B})$  is equivalent to projecting onto each half-space separately. Consider such a half-space corresponding to some  $\gamma$ , defined by  $a^\top x \leq -\mathbb{I}(\gamma)$ , where  $a \in \{1, 2\}^p$ , for some  $p \in \mathbb{N}$ . The projection of some  $z \in \mathbb{R}^p$  onto this half-space is given by

$$\arg \min_{x \in \mathbb{R}^p: a^\top x \leq -\mathbb{I}(\gamma)} (x - z)^\top \text{Diag}(a)(x - z). \quad (5.15)$$

The presence of  $\text{Diag}(a)$  in the objective function ensures that off-diagonal elements of  $Y$  are weighted with a factor of 2, as they appear twice in  $Y$ . The following result shows that (5.15) can be expressed in closed form.

**Lemma 5.2.** *Let  $p \in \mathbb{N}$ ,  $a, z \in \mathbb{R}^p$  with  $a > 0$ ,  $b \in \mathbb{R}$ , and denote by  $\mathbf{1}_p \in \mathbb{R}^p$  the all-ones vector. We have that*

$$\arg \min_{x \in \mathbb{R}^p: a^\top x \leq b} (x - z)^\top \text{Diag}(a)(x - z) = z - \frac{\max\{a^\top z - b, 0\}}{\mathbf{1}_p^\top a} \mathbf{1}_p. \quad (5.16)$$

*Proof.* Let  $f(x) := (x - z)^\top \text{Diag}(a)(x - z)$ . We need to determine the minimizer of the problem

$$\min_{x \in \mathbb{R}^p} f(x) \text{ subject to } a^\top x \leq b. \quad (5.17)$$

We consider two cases. If  $a^\top z \leq b$ , then  $z$  minimizes (5.17), since for any  $x \in \mathbb{R}^p$ , we have  $0 = f(z) \leq f(x)$ . Here, the inequality follows from the fact that  $a > 0$ , which makes  $\text{Diag}(a)$  positive definite.

If  $a^\top z > b$ , we consider the Karush-Kuhn-Tucker (KKT) conditions [159, 171] corresponding to (5.17), which state the following: if  $(x^*, \lambda^*)$ , with  $\lambda^* \geq 0$ , is a saddle point of the Lagrangian  $\mathcal{L}(x, \lambda) := f(x) + \lambda (a^\top x - b)$ , then  $x^*$  minimizes (5.17). The saddle point of  $\mathcal{L}(x, \lambda)$  is computed as follows:

$$\frac{\partial \mathcal{L}(x, \lambda)}{\partial x} = 2 \text{Diag}(a)(x - z) + \lambda a = 0 \implies x^* = z - \frac{\lambda^*}{2} \mathbf{1}_p.$$

Solving  $\partial \mathcal{L}(x^*, \lambda)/\partial \lambda = a^\top x^* - b = a^\top (z - \frac{\lambda^*}{2} \mathbf{1}_p) - b = 0$  for  $\lambda^*$  yields  $\lambda^* = 2(a^\top z - b)/(\mathbf{1}_p^\top a) \geq 0$ , where the inequality follows from  $a^\top z > b$ . Then  $x^* = z - \frac{a^\top z - b}{\mathbf{1}_p^\top a} \mathbf{1}_p$  minimizes (5.17) by the KKT conditions.

The proof follows by combining the two cases into the form (5.16).  $\square$

### 5.3.3 Upper bounds on $\alpha(G)$ from the ADMM iterates

The following lemma shows that any matrix in  $\mathcal{S}_+^{|\mathcal{B}|}$  induces an upper bound on  $\alpha(G)$ . Note that the matrix  $X^\ell$  from the ADMM iterates (5.12) satisfies  $X^\ell \in \mathcal{S}_+^{|\mathcal{B}|}$  for all  $\ell \in \mathbb{N}$ . Thus, any iteration of the ADMM provides an upper bound on  $\alpha(G)$ .

**Lemma 5.3.** *Let  $G = (V, E)$ , where  $V = [n]$  for some  $n \in \mathbb{N}$ ,  $\binom{[n]}{\leq 1} \subseteq \mathcal{B} \subseteq \binom{[n]}{\leq n}$ ,  $M \in \mathcal{S}_+^{|\mathcal{B}|}$ , and let  $(f_\beta)_{\beta \in \mathcal{B}^{2\cup} \cap [n]_G}$  satisfy  $(\mathbf{x}^\mathcal{B})^\top M \mathbf{x}^\mathcal{B} \equiv \sum_{\beta \in \mathcal{B}^{2\cup} \cap [n]_G} f_\beta x^\beta \pmod{\mathcal{I}_G}$ . We have that the value*

$$v(M) := M_{\emptyset, \emptyset} + \sum_{\beta \in \mathcal{B}^{2\cup} \cap [n]_G: \beta \neq \emptyset} \max \{f_\beta + \mathbb{I}(\beta), 0\} \quad (5.18)$$

satisfies  $v(M) \geq \alpha(G)$ .

*Proof.* For notational convenience, we define  $\mathcal{B}' := \mathcal{B}^{2\cup} \cap [n]_G$ . Consider the polynomial

$$\begin{aligned} g(x) &:= (\mathbf{x}^\mathcal{B})^\top M \mathbf{x}^\mathcal{B} + \sum_{\beta \in \mathcal{B}': \beta \neq \emptyset} \max \{f_\beta + \mathbb{I}(\beta), 0\} (1 - x^\beta) \\ &\quad + \sum_{\beta \in \mathcal{B}': \beta \neq \emptyset} \max \{-f_\beta - \mathbb{I}(\beta), 0\} x^\beta. \end{aligned}$$

Observe that for any  $x \in S_G$ ,  $g(x) \geq 0$  since  $M \succeq 0$ , and  $1 - x^\beta, x^\beta \geq 0$ . Hence,  $g \in \mathbb{P}_G$ .

Let  $(g_\beta)_{\beta \in \mathcal{B}'}$  be the coefficients of  $g$ , i.e.,  $g \equiv \sum_{\beta \in \mathcal{B}'} g_\beta x^\beta \pmod{\mathcal{I}_G}$ . For nonempty  $\beta \in \mathcal{B}'$ , we have

$$g_\beta = f_\beta - \max \{f_\beta + \mathbb{I}(\beta), 0\} + \max \{-f_\beta - \mathbb{I}(\beta), 0\} = -\mathbb{I}(\beta).$$

Thus,  $g(x) \equiv g_\emptyset - \sum_{i \in [n]} x_i \pmod{\mathcal{I}_G}$ , where  $g_\emptyset$  is the constant term of  $g$ , given by  $g_\emptyset = M_{\emptyset, \emptyset} + \sum_{\beta \in \mathcal{B}': \beta \neq \emptyset} \max \{f_\beta + \mathbb{I}(\beta), 0\}$ . By (5.2) and the fact that  $g \in \mathbb{P}_G$ , it follows that  $g_\emptyset \geq \alpha(G)$ . Since  $g_\emptyset = v(M)$ , this proves the result.  $\square$

Thus, it follows from Lemma 5.3 that for  $X^\ell$  as in (5.12) and  $v$  as in (5.18),  $v(X^\ell)$  is an upper bound on the stability number  $\alpha(G)$ . We will use the value  $v(X^\ell)$  as valid upper bound on  $\alpha(G)$  for the numerical results in Section 5.5.

## 5.4 A dynamic basis selection method and ADMM initialization

We provide a method for selecting a basis  $\binom{[n]}{\leq 1} \subseteq \mathcal{B} \subseteq \binom{[n]}{\leq 2}$  for computing  $\alpha^{\mathcal{B}}(G)$ , see (5.5). We also provide a method for initializing the ADMM iterates  $X^1$ ,  $Y^1$ ,  $Z^1$ , see (5.12). Both these methods are based on an SDP defining the Lovász theta function of a graph  $G = ([n], E)$ , with  $n \in \mathbb{N}$ . This SDP is given by:

$$\begin{aligned} \vartheta(G) = \max_{Z \in \mathcal{S}_+^{1+n}} \quad & \sum_{i \in [n]} Z_{\emptyset, i} \\ \text{s.t.} \quad & Z_{i, j} = 0 \text{ for all } \{i, j\} \in E \\ & Z_{i, i} = Z_{\emptyset, i} \text{ for all } i \in [n], \quad Z_{\emptyset, \emptyset} = 1. \end{aligned} \tag{5.19}$$

Here, the matrix  $Z$  is indexed by the  $1 + n$  elements of  $\binom{[n]}{\leq 1}$  (see also [120, Sect. 3.1]). It can be shown that (5.19) is dual to the SDP defining  $\alpha^{\binom{[n]}{\leq 1}}(G)$ , see (5.8). To solve (5.19), we use the IPM-based SDP solver MOSEK [228]. The PSD variable in (5.19) is of order  $1 + n$  and there are  $2|E| + 2n + 1$  linear equality constraints, which is small enough for IPM-based solvers to be efficient for graphs of sizes considered here. For instance, among the graphs we consider, `c_fat200_5` attains the largest value of  $2|E| + 2n + 1$ , with  $|E| = 11427$  and  $n = 200$  (see Table 5.7). For this graph, MOSEK solves (5.19) in approximately 35 seconds.

### 5.4.1 The basis selection method

Given a graph  $G$  on  $n$  vertices, and a maximum basis size  $s \geq 1 + n$ , our method aims to select a basis  $\binom{[n]}{\leq 1} \subseteq \mathcal{B} \subseteq \binom{[n]}{\leq 2}$ ,  $|\mathcal{B}| \leq s$  that minimizes  $\alpha^{\mathcal{B}}(G)$ . The inclusions  $\binom{[n]}{\leq 1} \subseteq \mathcal{B} \subseteq \binom{[n]}{\leq 2}$  can be interpreted in terms of  $\mathcal{F}(\mathcal{B})$ , see (5.9), as follows: matrices in  $\mathcal{F}\left(\binom{[n]}{\leq 1}\right)$  are submatrices of matrices in  $\mathcal{F}(\mathcal{B})$ , which in turn are submatrices of matrices in  $\mathcal{F}\left(\binom{[n]}{\leq 2}\right)$ .

Recall that the first and second levels of the Lasserre hierarchy for the stable set problem are the SDPs defining  $\alpha^{\binom{[n]}{\leq 1}}(G)$  and  $\alpha^{\binom{[n]}{\leq 2}}(G)$  respectively. Thus, our method selects a basis  $\mathcal{B}$  such that  $\alpha^{\mathcal{B}}(G)$  corresponds to a level of the Lasserre hierarchy intermediate to levels 1 and 2, and for some smaller graphs, equal to level 2. We do not consider bases corresponding to levels  $k > 2$ , since the value  $\alpha^{\binom{[n]}{\leq 2}}(G)$ , after rounding down to the nearest integer, often closes the gap with  $\alpha(G)$ , or is intractable to compute.

To explain our method, we define, for any subset  $\beta \subseteq [n]$ , the binary matrix  $Z^\beta := \begin{bmatrix} \mathbf{1} \\ \mathbb{1}_\beta \end{bmatrix} \begin{bmatrix} \mathbf{1} \\ \mathbb{1}_\beta \end{bmatrix}^\top$ . Matrix  $Z^\beta$  is indexed by the  $1 + n$  elements of  $\binom{[n]}{\leq 1}$ . Observe that  $Z^\beta$  is feasible for (5.19) if and only if  $\beta \in [n]_G$ . Let  $\beta \in [n]_G$  correspond to a maximum stable set, and consider the binary values  $(Z_{i, j}^\beta)_{\{i, j\} \notin E}$ . Observe that  $Z_{i, j}^\beta = 1$  if and only if both vertices  $i, j \in \beta$ . Thus, the values 1 in the vector  $(Z_{i, j}^\beta)_{\{i, j\} \notin E}$  indicate

the maximum stable set  $\beta$ . As such, we would like to include the monomials  $x_i x_j$ , for which  $Z_{i,j}^\beta = 1$ , in our basis. In practice however, the matrix  $Z_{i,j}^\beta$  is unknown, since a maximum stable set is not known. Therefore, instead of  $Z^\beta$ , we consider matrix  $Z^*$ , which denotes an optimal solution of (5.19), and can be considered a semidefinite approximation of  $Z^\beta$ . Then, we include monomial  $x_i x_j$ ,  $\{i, j\} \notin E$ , in our basis if  $Z_{i,j}^*$  is ‘large enough’.

We now present our basis selection method formally: Compute first  $s'$ , defined as the cardinality of  $\binom{[n]}{\leq 2} \setminus \left( \binom{[n]}{\leq 1} \cup E \right)$ , which is the set of non-edges in  $G$ . Then we distinguish two cases, based on the given maximum basis size  $s$ .

**Case 1:**  $s < 1 + n + s'$ . Solve (5.19) using an IPM-based SDP solver to obtain an optimal solution  $Z^*$ . As basis  $\mathcal{B}$ , we choose the sets in  $\binom{[n]}{\leq 1}$  and the  $s - (1 + n)$  non-edges  $\{i, j\}$  with largest value  $Z_{i,j}^*$ . Note that  $|\mathcal{B}| = \left| \binom{[n]}{\leq 1} \right| + s - (1 + n) = s$ .

**Case 2:**  $s \geq 1 + n + s'$ . We take the basis  $\mathcal{B} = \binom{[n]}{\leq 2} \setminus E$ , of size  $|\mathcal{B}| = 1 + n + s' \leq s$ . Observe that  $\mathcal{B} = \binom{[n]}{\leq 2} \cap [n]_G$ , which implies that  $\alpha^{\mathcal{B}}(G) = \alpha^{\binom{[n]}{\leq 2}}(G)$ , see Lemma 5.1. Therefore, in Case 2, the resulting basis corresponds to the second level of the Lasserre hierarchy.

Preliminary numerical experiments showed that selecting monomials based on the largest values of  $(Z_{i,j}^*)_{\{i,j\} \notin E}$ , outperformed methods that selected monomials based on smallest values in  $(Z_{i,j}^*)_{\{i,j\} \notin E}$ , or those values closest to the average value given by  $\sum_{\{i,j\} \in E} Z_{i,j}^* / |E|$ . We also tested basis selection methods based on the values  $(Z_{\emptyset,i}^*)_{i \in [n]}$ , or degrees of vertices, and these were also outperformed by the above described basis selection method.

## 5.4.2 Initialization of ADMM iterates

Our initialization method is defined for any basis  $\binom{[n]}{\leq 1} \subseteq \mathcal{B} \subseteq \binom{[n]}{\leq n}$ , and depends on optimal primal and dual solutions to (5.19), denoted respectively by  $Z^* \in \mathcal{S}_+^{1+n}$  and  $X^* \in \mathcal{S}_+^{1+n}$ .

Note that the initial ADMM iterates  $X^1, Y^1, Z^1$  are indexed by the elements of  $\mathcal{B}$ . Our initialization method sets the principal submatrix of  $X^1$ , that is indexed by the elements of  $\binom{[n]}{\leq 1}$ , equal to  $X^*$ . We set the other elements of  $X^1$  to zero, and set  $Y^1 = X^1$ . We initialize  $Z^1$  by setting the principal submatrix of  $Z^1$  corresponding to the elements of  $\binom{[n]}{\leq 1}$  as  $(1/\rho)Z^*$ , and the rest all zero. Here, we scale  $Z^*$  by  $1/\rho$ , since (5.12) corresponds to the scaled form of the ADMM.

An important property of this initialization is that  $v(X^1)$ , see (5.18), equals  $\vartheta(G)$ . Indeed, since  $X^* \succeq 0$ , also  $X^1 \succeq 0$ . Moreover, since  $X^*$  is a feasible dual solution to (5.19), and (5.19) is the SDP dual of (5.8) for  $\mathcal{B} = \binom{[n]}{\leq 1}$ , it follows that  $X^* \in \mathcal{F}\left(\binom{[n]}{\leq 1}\right)$ . This implies that the values  $f_\beta$  in (5.18), corresponding to  $X^1$ , satisfy  $f_\beta \leq -\mathbb{I}(\beta)$ , from where it follows that  $v(X^1) = X_{\emptyset,\emptyset}^1$ . Lastly, since  $X^*$  is an optimal dual solution to (5.19),  $X_{\emptyset,\emptyset}^* = X_{\emptyset,\emptyset}^1 = \vartheta(G)$ . Thus,  $v(X^1) = \vartheta(G)$ . Consequently, the best bound returned by the ADMM after a finite number of iterations is at most  $\vartheta(G)$ .

## 5.5 Numerical results

We compute Lasserre hierarchy bounds for the stable set problem, using the ADMM as described in Section 5.3.2, with stepsize parameter  $\nu = 3/2$ , see (5.12). We set the ADMM penalty parameter  $\rho = (4/5)\sqrt{|\mathcal{B}|}$ , see (5.11), where  $\mathcal{B}$  is the used basis of size at most  $s \in \mathbb{N}$ , determined by the basis selection method outlined in Section 5.4.1. The algorithms are implemented in MATLAB. All experiments are run on a machine with 16GB RAM and an Intel i7-1165G7 CPU.

In Section 5.5.1 we compare SDP bounds obtained from two versions of the ADMM algorithm: one that computes the eigendecomposition in (5.14) using single precision, and another that uses double precision. We conclude that the single precision ADMM requires less computation time per iteration, without a significant loss in quality of the corresponding bounds. The stopping conditions are provided in Section 5.5.2, and are used in Section 5.5.3 to compute bounds on  $\alpha(G)$  from the Lasserre hierarchy with the single precision ADMM. We compare these bounds with the bounds obtained by the exact subgraph hierarchy (ESH) [98] and the bounds from [260]. Both approaches provide among the strongest SDP bounds for the stable set problem.

### 5.5.1 Single vs. double precision for eigendecompositions

It is well known that one of the most expensive steps of the ADMM, when applied to SDP, is computing projections onto the PSD cone, see e.g., [243, Sect. 5] and [266, Sect. 1]. It is standard to compute these projections  $\mathcal{P}_{\mathcal{S}_+^{|\mathcal{B}|}}(A)$ , see (5.14), by computing the full eigendecomposition of  $A$ , that is,  $A = \sum_{\lambda \in \Lambda(A)} \lambda u_\lambda u_\lambda^\top$ . Computing the eigendecomposition in single precision is computationally less expensive than in double precision. However, the lower accuracy of single precision might result in worse upper bounds compared to double precision. We investigate this trade-off by comparing the SDP bounds on the stable set problem obtained by two versions of the ADMM algorithm: one that uses single precision and another that uses double precision for the eigendecompositions. All other parts of the algorithms remain the same.

We run each version of the ADMM algorithm for a fixed number of iterations to compute bounds on  $\alpha(G)$  for three different graphs. All ADMM iterates, see (5.12), are initialized by setting  $X^1 = Y^1 = Z^1 = 0$ , i.e., the zero matrix of appropriate size. When the iteration number  $\ell$  is a multiple of 100, we compute the bound  $v(X^\ell)$ , see (5.18), which satisfies  $v(X^\ell) \geq \alpha(G)$ . We also track the computation time, and present the results in Tables 5.1 to 5.3. In these tables, column ‘Bnd. diff.’ (for bound difference) reports the double precision bound  $v(X^\ell)$  subtracted from the single precision  $v(X^\ell)$ .

From the data presented in these tables we conclude that the difference in  $v(X^\ell)$  between the two precisions and for fixed  $\ell$ , is negligible (at most 0.04517). In contrast, the reduction in computation time may be significant, see for instance the row corresponding to  $\ell = 600$  in Table 5.3: the single precision ADMM requires 113.82 seconds to compute 600 iterations, whereas the double precision ADMM requires 205.35 seconds. With this larger computation time, the double precision bound  $v(X^\ell)$  is only

0.00677 smaller than the single precision bound. Rounded down, both precisions provide the same bound on  $\alpha(G)$ , but the single precision ADMM requires only 55% of the computation time of the double precision ADMM.

We also perform the following similar experiment: we run each ADMM version for one hour on a single graph. We pick a basis of size  $s = 2500$  and initialize the ADMM iterates with the methods outlined in Sections 5.4.1 and 5.4.2, respectively. At the first iteration and every 10 seconds  $t$ , we compute the valid upper bound  $v_t := v(X^\ell)$ , where  $\ell$  is the iteration index at time  $t$ . We set  $v_0 := v(X^1) = \vartheta(G)$ . This yields a set of upper bounds  $\{v_0, v_{10}, \dots, v_{3600}\}$ . Figure 5.1 reports the best bounds achieved by both ADMM versions over time. That is, Figure 5.1 plots the curves through the points  $(t, \min_{k \in \{0, 10, \dots, t\}} v_k)$ , for  $t \in \{0, 10, 20, \dots, 3600\}$ . For the single precision version of the ADMM algorithm, the bounds  $v_{10}, v_{20}, \dots, v_{80} \geq v_0$ , and thus the plot corresponding to single precision is flat for the first 80 seconds. For the double precision variant of the ADMM, the plot is flat for the first 170 seconds. Figure 5.1 demonstrates that the single precision ADMM provides stronger bounds than the double precision ADMM, at any time  $t$ ,  $t \leq 3600$ . This is due to the fact that the single precision ADMM algorithm can perform more ADMM iterations in the same time as compared to the double precision ADMM algorithm.

Based on these conclusions, we will use the version of the ADMM that computes eigendecompositions in single precision for the remainder of this section.

## 5.5.2 Stopping conditions

For each graph, we run the ADMM algorithm for at most one hour (unless otherwise specified). Next to maximum running time, we have the following stopping conditions: we stop if

$$\max \left\{ \frac{\|X^\ell - Y^\ell\|}{1 + \|X^\ell\|}, \rho \frac{\|X^{\ell-1} - X^\ell\|}{1 + \|X^\ell\|} \right\} \leq 10^{-4}, \quad (5.20)$$

for 3 consecutive iterations, see e.g., [34, Sect. 3.3.1]. To keep computation costs minimal, we only verify whether (5.20) holds whenever the iteration number  $\ell$  is a multiple of 100 (and then also for the consecutive iterations if (5.20) holds). We also stop earlier if the objective value  $X_{\emptyset, \emptyset}^\ell$  stagnates. Specifically, we stop if

$$\left| X_{\emptyset, \emptyset}^\ell - X_{\emptyset, \emptyset}^{\ell-1} \right| < 10^{-5} \quad (5.21)$$

for  $K_{\text{stag}} \in \mathbb{N}$  (not necessarily consecutive) iterations. For all the tables that follow, except Table 5.5, we set  $K_{\text{stag}} = 150$ .

## 5.5.3 SDP-Lasserre bounds on $\alpha(G)$

We investigate the quality of the upper bounds on the stability number of graphs obtained from the Lasserre hierarchy, and computed via the ADMM. For the remainder of this section, we refer to such bounds as SDP-Lasserre bounds. To ensure that these bounds are valid, we use Lemma 5.3. For all the tables that follow, except

$\ell$	Single precision		Double precision		Bnd. diff.
	$v(X^\ell)$	Time (s)	$v(X^\ell)$	Time (s)	
100	7.13476	0.14	7.13484	0.16	-0.00008
200	7.03090	0.25	7.03090	0.32	0.00000
300	7.07006	0.36	7.07013	0.47	-0.00007
400	7.00445	0.46	7.00444	0.61	0.00000
500	7.00680	0.57	7.00677	0.76	0.00004
600	7.00032	0.68	7.00032	0.91	0.00001

Table 5.1: Comparison of upper bounds for HoG\_15599,  $n = 20$ ,  $\alpha(G) = 7$ ,  $s = 126$ .

$\ell$	Single precision		Double precision		Bnd. diff.
	$v(X^\ell)$	Time (s)	$v(X^\ell)$	Time (s)	
100	20.92761	5.14	20.92756	7.99	0.00005
200	23.01029	9.97	23.01063	16.05	-0.00035
300	17.21289	15.28	17.21270	24.45	0.00019
400	16.42331	21.03	16.42139	33.03	0.00192
500	16.51678	26.66	16.51551	41.41	0.00127
600	16.30446	32.15	16.30274	50.83	0.00172
700	16.49339	37.60	16.49267	59.29	0.00072
800	16.29119	43.19	16.28875	67.88	0.00245
900	16.39126	49.01	16.38704	76.83	0.00421
1000	16.28313	54.91	16.27664	85.77	0.00649
1100	16.31098	62.01	16.30117	94.50	0.00981

Table 5.2: Comparison of upper bounds for MANN\_a9\_c1q,  $n = 45$ ,  $\alpha(G) = 16$ ,  $s = 964$ .

Table 5.5, we set the maximum basis size  $s = 2500$ . This value is chosen to ensure the ADMM converges within one hour on most graphs. In the remainder of this section, we initialize the ADMM iterates using the method described in Section 5.4.2.

### 5.5.3.1 SDP bounds for evil, random and near-regular graphs

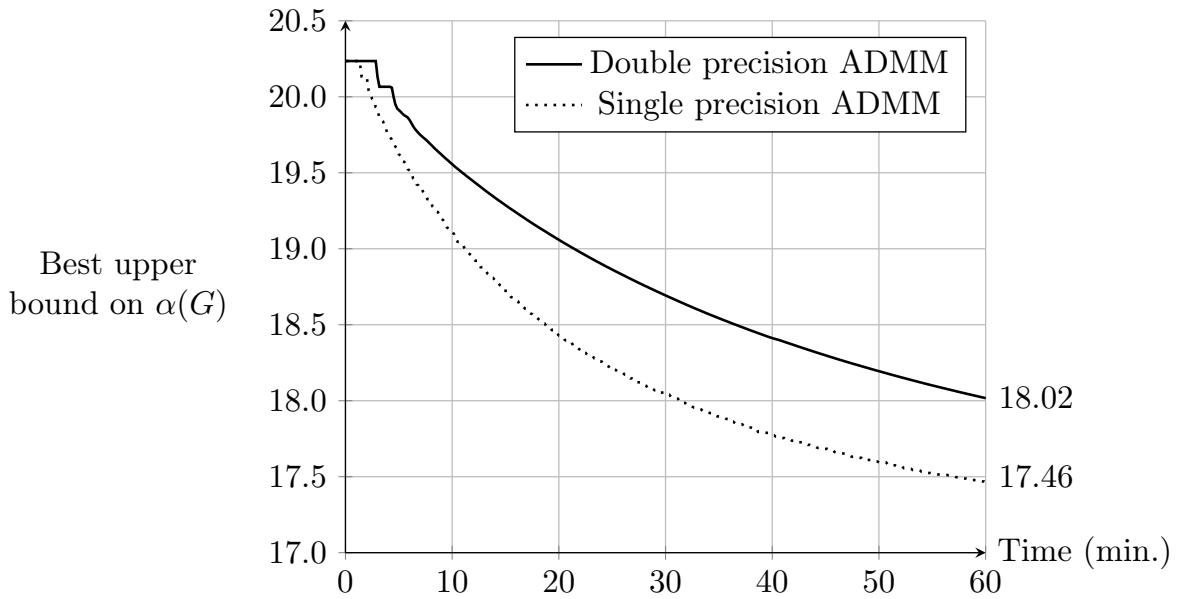
We benchmark the SDP-Lasserre bounds on the complement of evil graphs<sup>1</sup> [292], as well as on random graphs and near-regular graphs. These graphs are also tested in [260], and can be described as follows:

- **Evil graphs [292].** These are benchmark graphs for the NP-hard clique problem. The clique problem on a graph  $G$  is equivalent to the stable set problem on the complement graph of  $G$ . Therefore, we consider the complement of evil graphs.

The name of all evil graphs we consider starts with the prefix `evil-N[n]-p98`, where  $n$  is the number of vertices of the corresponding evil graph. To fit the

<sup>1</sup>The evil graphs are available at <https://github.com/zbogdan/evil2/tree/main/evil-tests>.

$\ell$	Single precision		Double precision		Bnd. diff.
	$v(X^\ell)$	Time (s)	$v(X^\ell)$	Time (s)	
100	122.53210	19.66	122.52626	35.47	0.00584
200	82.37548	41.58	82.37179	70.49	0.00368
300	78.23423	59.13	78.22998	105.84	0.00425
400	78.89802	76.84	78.86581	139.59	0.03221
500	71.93651	96.10	71.89134	173.73	0.04517
600	69.47637	113.82	69.46960	205.35	0.00677

Table 5.3: Comparison of upper bounds for E-myc5x30,  $n = 150$ ,  $\alpha(G) = 60$ ,  $s = 1500$ .Figure 5.1: Comparison of upper bounds for E-myc23x8,  $n = 184$ ,  $\alpha(G) = 16$ ,  $s = 2500$ .

tables on one page, we replace this prefix by simply E. For example, the evil graph evil-N120-p98-chv12x10 will be reported as E-chv12x10.

- **Random graphs.** These are generated following the Erdős-Rényi model. That is, given  $n \in \mathbb{N}$  and  $p \in (0, 1)$ , generate a graph by taking  $n$  vertices and creating edges  $\{i, j\}$  independently with probability  $p$  for every  $i, j \in [n]$ .

In the tables, we report the random graphs as `rnd_p[p]`, where  $p$  is the corresponding probability. For example, `rnd_p004` is the random graph with  $p = 0.04$ , and  $n$  can be read from the table.

- **Near-regular graphs.** For given  $n, r \in \mathbb{N}$  such that  $nr$  is even, these graphs are constructed as follows (see also [98, Sect. 7.2]): consider a set of  $nr$  vertices given by  $\tilde{V} := \{\{i, b\} : i \in [n], b \in [r]\}$ . Select a perfect matching on the vertices in  $\tilde{V}$  to obtain the edge set  $\tilde{E} \subseteq \tilde{V} \times \tilde{V}$ . Consider now the graph  $G$  with vertices  $V = [n]$  and edge set  $E = \{\{i, j\} : \exists b, b' \in [r] \text{ s.t. } \{\{i, b\}, \{j, b'\}\} \in \tilde{E}\}$ . Note

that  $G$  is a regular multigraph. Remove from  $G$  any parallel edges and self-loops. The resulting graph is said to be near-regular.

In the tables, we report near-regular graphs as `reg_r[r]`, where  $r$  is the parameter that was just described. The number of vertices can be read from the table.

We use the same exact random and near-regular graphs as in [260]. For each graph, we use the method provided in Section 5.4.1 to select a basis of size 2500 for computing the SDP-Lasserre bounds. Because  $2500 < \left| \binom{[n]}{\leq 2} \cap [n]_G \right|$  for all graphs, the resulting bounds correspond to the Lasserre hierarchy at levels intermediate to 1 and 2.

Table 5.4 reports the SDP-Lasserre bounds on  $\alpha(G)$ , computed via the ADMM, in the column ‘SDP-Lasserre’. Columns  $n$ ,  $|E|$ , and  $\vartheta(G)$  report the number of vertices, number of edges, and Lovász theta function respectively, for each graph. Column  $\alpha$  reports (bounds on) the stability number of the graphs. For evil graphs, the value of  $\alpha(G)$  is known by construction. The values and intervals for the stability numbers of the random and near-regular graphs are taken from [99, Table 6]. In Table 5.5, we present improved bounds on  $\alpha(G)$  compared to [99, Table 6].

The columns under ‘BOUND 2 [260]’ in Table 5.4 report BOUND 2 from the recent paper [260, Sect. 4.1], which provides one of the best SDP-based upper bounds on  $\alpha(G)$ . BOUND 2 is obtained by strengthening the Lovász theta function with additional valid inequalities, such as triangle inequalities and inequalities induced by complete subgraphs of  $G$  (i.e., constraints of the form  $\sum_{i \in U} x_i \leq 1$  where  $U$  is a set of pairwise connected vertices). BOUND 2 is computed by the IPM-based SDP solver MOSEK [228]. We computed BOUND 2 on the same machine we used to compute the SDP-Lasserre bounds<sup>2</sup>. Additional details regarding BOUND 2 can be found in [260, Sect. 4.1].

The columns ‘GapClsd’ (for gap closed) under SDP-Lasserre and BOUND 2 report the fraction  $\frac{\vartheta(G) - f^*}{\vartheta(G) - \alpha(G)}$  rounded down to the first digit, where  $f^*$  equals the value of the corresponding bound. Note that  $f^* \in [\alpha(G), \vartheta(G)]$ . For the three graphs with unknown  $\alpha(G)$ , we use the best known lower bound on  $\alpha(G)$  from [99, Table 6] to compute GapClsd. Lastly, bounds that equal  $\alpha(G)$  when rounded down are boldfaced.

Computing BOUND 2 for the graphs in Table 5.4 requires on average only 4 minutes of computation time per graph. The ADMM required on average 39 minutes per graph. The SDP-Lasserre bounds improve over BOUND 2 for various graphs, sometimes closing the gap towards  $\alpha(G)$  whereas BOUND 2 did not (see the graphs `E-s3m25x5`, `E-myc23x6`, `reg_r6` and `reg_r8`). There is one graph in Table 5.4, `rnd_p002`, for which the SDP-Lasserre bound does not improve upon the Lovász theta function  $\vartheta(G) = \alpha^{\binom{[n]}{\leq 1}}(G) = 95.778$ . We ran our ADMM algorithm for 4 hours to recompute the SDP-Lasserre bound for this graph, which resulted in an improved bound of 95.244, see also Table 5.8.

We rerun the ADMM algorithm on the graphs `reg_r10` and `reg_r6` with increased basis size  $|\mathcal{B}|$  and extended running time. These changes yield improved upper bounds on  $\alpha(G)$  compared to those in [99, Table 6], see also Table 5.4. The improved bounds are reported in Table 5.5, where column ‘ $|\mathcal{B}|$ ’ reports the used basis size and column

<sup>2</sup>The code for BOUND 2 is available at <https://arxiv.org/src/2401.17069v2/anc>.

‘T. (min.)’ reports the runtime rounded to the nearest minute. In these computations, we use  $K_{\text{stag}} = 500$ , see (5.21).

Lastly, we compare the SDP-Lasserre bounds also with the ESH approach from [98]. Specifically, we compare the SDP-Lasserre bounds with the ESH bounds on the random and near-regular graphs reported in Tables 4 and 5 of [98]. The results are presented in Table 5.6, with identical columns as Table 5.4, and the newly added column ‘ESH’. The columns under ‘ESH’ report the obtained bounds by the ESH and corresponding GapClsd values. The SDP-Lasserre bounds improve over the ESH bounds in 11 of the 15 reported graphs. The computation times of these ESH bounds is reported in [98], although the used computer is not specified. For each of the graphs in Table 5.6, the ESH required on average 1454 seconds. Note that SDP-Lasserre bounds, when rounded down to the closest integer, close the gap for more graphs than the other two approaches.

### 5.5.3.2 SDP bounds on graphs from [97]

We investigate the quality of the SDP-Lasserre bounds on the graphs tested in [97]. This set of graphs<sup>3</sup> contains, among others, complements of DIMACS graphs, additional random Erdős-Rényi graphs, a spin glass graph, graphs from [41], as well as a Paley, a circulant and a cubic graph. Note that stability numbers of the DIMACS graphs are known, see e.g., [23].

For all graphs, we use the method from Section 5.4.1 to select a basis of size at most 2500 for the ADMM. The tested graphs on  $n \leq 80$  vertices satisfy  $\left| \binom{[n]}{\leq 2} \cap [n]_G \right| \leq 2500$ , which implies that we compute the full second level of the Lasserre hierarchy for those graphs. The results are reported in Table 5.7, with the same column definitions as in Table 5.4, except for the new column denoted by  $|\mathcal{B}|$ . This column reports the size of the used basis (in Table 5.4,  $|\mathcal{B}| = 2500$  for every graph). Bounds that equal  $\alpha(G)$  when rounded down are boldfaced.

We again compare our SDP-Lasserre bounds with BOUND 2 from [260]. Both these bounds are stronger than the bounds in [97]. Computing BOUND 2 for the graphs in Table 5.7 requires on average only two minutes of computation time per graph. However, for the larger graphs ( $n \geq 150$ ), BOUND 2 cannot be computed due to insufficient memory, as indicated by the table entry ‘-’. The large memory requirement of BOUND 2 is due to the large number of linear constraints. The SDP-Lasserre bounds improve significantly over  $\vartheta(G)$ , and often also over BOUND 2. In particular, the SDP-Lasserre bounds often close the gap towards  $\alpha(G)$  when rounded down. All graphs with  $n < 72$  took at most 122 seconds, except for G\_60\_025 which required approximately 270 seconds. For graphs with  $n \geq 72$ , the ADMM algorithm required between 15 and 60 minutes to terminate.

For the graph c\_fat200\_5 in Table 5.7, the SDP-Lasserre bound does not improve over  $\vartheta(G) = 60.345$ . We ran our ADMM algorithm for 4 hours to recompute the SDP-Lasserre bound of this graph, which resulted in an improved bound of 60.317, see also Table 5.8.

---

<sup>3</sup>The graphs used in [97] are available at [https://arxiv.org/src/2003.13605v6/anc/Data\\_InputGraphs.mat](https://arxiv.org/src/2003.13605v6/anc/Data_InputGraphs.mat).

### 5.5.3.3 SDP bounds on SageMath graphs

We compute the SDP-Lasserre bounds for several graphs from the SageMath [295] software. Specifically, we consider SageMath graphs on at least 30 vertices, for which  $\vartheta(G)$  is strictly larger than  $\alpha(G)$ .

The results are reported in Table 5.9, which uses the same column definitions as Table 5.7. The SDP-Lasserre bounds improve significantly over  $\vartheta(G)$ , and all of them equal  $\alpha(G)$  when rounded down. On average, the ADMM algorithm required 1098 seconds to terminate for each graph. It is worth noting that for each graph, the ADMM required at most 900 seconds to reach an iteration  $\ell$  for which  $\lfloor v(X^\ell) \rfloor = \alpha(G)$ , see (5.18). The computation of BOUND 2 required at most 220 seconds, but there are some graphs in Table 5.9 for which the floor of BOUND 2 does not equal  $\alpha(G)$ .

Graph name	$n$	$ E $	$\alpha$	SDP-Lasserre		BOUND 2 [260]		$\vartheta(G)$
				Bound	GapClsd	Bound	GapClsd	
E-chv12x10	120	545	20	<b>20.355</b>	92.1%	<b>20.000</b>	99.9%	24.526
E-myc5x24	120	236	48	<b>48.466</b>	89.8%	<b>48.000</b>	99.9%	52.607
E-myc11x11	121	508	22	<b>22.361</b>	91.7%	<b>22.000</b>	99.9%	26.397
E-s3m25x5	125	873	20	<b>20.291</b>	94.1%	22.361	52.7%	25.000
E-myc23x6	138	1242	12	<b>12.501</b>	84.2%	15.177	0.0%	15.177
E-myc5x30	150	338	60	61.477	71.1%	<b>60.000</b>	99.9%	65.121
E-s3m25x6	150	1102	24	25.239	79.3%	26.833	52.7%	30.000
E-myc11x14	154	701	28	<b>28.316</b>	94.3%	<b>28.000</b>	99.9%	33.596
E-chv12x15	180	944	30	<b>30.391</b>	94.2%	<b>30.000</b>	99.9%	36.788
E-myc5x36	180	439	72	75.626	29.0%	<b>72.000</b>	99.9%	77.110
E-myc23x8	184	1764	16	17.504	64.4%	20.235	0.0%	20.235
E-myc11x17	187	901	34	<b>34.332</b>	95.1%	<b>34.000</b>	99.9%	40.795
E-s3m25x8	200	1550	32	35.979	50.2%	35.777	52.7%	40.000
E-myc5x42	210	541	84	86.513	58.1%	<b>84.000</b>	99.9%	90.001
E-myc11x20	220	1130	40	42.791	65.0%	<b>40.000</b>	99.9%	47.994
E-myc23x10	230	2263	20	22.553	51.7%	25.294	0.0%	25.294
E-chv12x20	240	1352	40	<b>40.461</b>	94.9%	<b>40.000</b>	99.9%	49.051
E-myc5x48	240	718	96	98.627	44.0%	<b>96.011</b>	99.7%	100.696
E-s3m25x10	250	2050	40	45.003	49.9%	44.721	52.7%	50.000
E-myc11x23	253	1456	46	49.447	62.5%	<b>46.014</b>	99.8%	55.193
E-myc5x60	300	1033	120	121.876	34.4%	<b>120.020</b>	99.2%	122.861
rnd_p004	100	212	45	<b>45.140</b>	86.8%	<b>45.027</b>	97.4%	46.067
rnd_p006	100	303	38	<b>38.199</b>	91.5%	<b>38.435</b>	81.5%	40.361
rnd_p008	100	443	32	<b>32.475</b>	83.3%	<b>32.433</b>	84.7%	34.847
rnd_p010	100	489	32	<b>32.029</b>	98.5%	<b>32.151</b>	92.5%	34.020
rnd_p002	200	407	95	<b>95.778</b>	0.0%	<b>95.043</b>	94.5%	<b>95.778</b>
rnd_p003	200	631	81	82.425	46.4%	<b>81.079</b>	97.0%	83.662
rnd_p004	200	816	67	69.890	58.1%	69.818	59.2%	73.908
rnd_p005	200	991	64	67.355	33.4%	65.544	69.3%	69.039
reg_r10	100	474	28	29.636	56.9%	29.431	62.3%	31.797
reg_r4	100	195	40	<b>40.333</b>	90.3%	<b>40.713</b>	79.3%	43.449
reg_r6	100	294	34	<b>34.667</b>	82.5%	35.047	72.5%	37.815
reg_r8	100	377	31	<b>31.645</b>	81.4%	32.063	69.4%	34.480
reg_r10	200	980	[57, 59]	60.052	67.5%	62.695	39.5%	66.418
reg_r4	200	400	81	82.450	78.5%	82.246	81.5%	87.759
reg_r6	200	593	[69, 72]	72.685	64.1%	73.709	54.1%	79.276
reg_r8	200	792	[60, 63]	64.749	55.9%	66.789	37.0%	70.790

Table 5.4: Results on evil, random and near-regular graphs.

Graph name	$n$	$ E $	$\alpha(G)$ bounds		SDP-Lasserre	$ \mathcal{B} $	T. (min.)
			New	Old [99]			
reg_r10	200	980	[57, 58]	[57, 59]	58.932	4250	180
reg_r6	200	593	[69, 71]	[69, 72]	71.821	3750	105

Table 5.5: Improved bounds on  $\alpha(G)$  with larger basis size  $|\mathcal{B}|$  and longer running time.

Graph name	$\alpha$	SDP-Lasserre		BOUND 2 [260]		ESH [98]		$\vartheta(G)$
		Bound	GapClsd	Bound	GapClsd	Bound	GapClsd	
rnd_p004	45	<b>45.140</b>	86.8%	<b>45.027</b>	97.4%	<b>45.021</b>	98.0%	46.067
rnd_p006	38	<b>38.199</b>	91.5%	<b>38.435</b>	81.5%	<b>38.439</b>	81.4%	40.361
rnd_p008	32	<b>32.475</b>	83.3%	<b>32.433</b>	84.7%	<b>32.579</b>	79.6%	34.847
rnd_p010	32	<b>32.029</b>	98.5%	<b>32.151</b>	92.5%	<b>32.191</b>	90.5%	34.020
rnd_p002	95	<b>95.778</b>	0.0%	<b>95.043</b>	94.5%	<b>95.032</b>	95.8%	<b>95.778</b>
rnd_p003	81	82.425	46.4%	<b>81.079</b>	97.0%	<b>81.224</b>	91.5%	83.662
rnd_p004	67	69.890	58.1%	69.818	59.2%	70.839	44.4%	73.908
rnd_p005	64	67.355	33.4%	65.544	69.3%	66.091	58.5%	69.039
reg_r4	40	<b>40.333</b>	90.3%	<b>40.713</b>	79.3%	<b>40.687</b>	80.0%	43.449
reg_r6	34	<b>34.667</b>	82.5%	35.047	72.5%	35.246	67.3%	37.815
reg_r8	31	<b>31.645</b>	81.4%	32.063	69.4%	32.190	65.8%	34.480
reg_r10	[57, 59]	60.052	67.5%	62.695	39.5%	62.894	37.4%	66.418
reg_r4	81	82.450	78.5%	82.246	81.5%	83.732	59.5%	87.759
reg_r6	[69, 72]	72.685	64.1%	73.709	54.1%	75.555	36.2%	79.276
reg_r8	[60, 63]	64.749	55.9%	66.789	37.0%	67.785	27.8%	70.790

Table 5.6: Comparison of the SDP-Lasserre, BOUND 2 and ESH bounds.

Graph name	$n$	$ E $	$\alpha$	SDP-Lasserre			BOUND 2 [260]		$\vartheta(G)$
				$ \mathcal{B} $	Bound	GapClsd	Bound	GapClsd	
HoG_34272	9	17	3	29	<b>3.000</b>	99.8%	<b>3.000</b>	99.9%	<b>3.338</b>
HoG_15599	20	44	7	167	<b>7.003</b>	99.6%	<b>7.000</b>	99.9%	<b>7.820</b>
CubicVT26_5	26	39	10	313	<b>10.100</b>	94.5%	<b>10.662</b>	63.5%	11.817
HoG_34274	36	72	12	595	<b>12.010</b>	99.1%	<b>12.000</b>	99.9%	13.232
HoG_6575	45	225	10	811	<b>10.132</b>	97.3%	13.220	36.2%	15.053
MANN_a9_clq	45	72	16	964	<b>16.281</b>	80.9%	17.225	16.9%	17.475
Circ47_030	47	282	13	847	<b>13.003</b>	99.7%	<b>13.026</b>	97.9%	14.302
G_50_025	50	308	12	968	<b>12.028</b>	98.2%	<b>12.367</b>	76.5%	13.564
G_60_025	60	450	13	1381	<b>13.003</b>	99.7%	<b>13.241</b>	81.1%	14.281
PaleyGraph61	61	915	5	977	<b>5.289</b>	89.7%	7.810	0.0%	7.810
hamming6_4	64	1312	4	769	<b>4.032</b>	97.5%	<b>4.749</b>	43.7%	5.333
HoG_34276	72	144	24	2485	<b>24.042</b>	98.2%	<b>24.000</b>	99.9%	26.463
G_80_050	80	1620	9	1621	<b>9.001</b>	99.7%	<b>9.092</b>	78.8%	<b>9.435</b>
G_100_025	100	1243	17	2500	<b>17.000</b>	99.9%	18.428	41.5%	19.441
spin5	125	375	50	2500	<b>50.236</b>	95.9%	<b>50.000</b>	99.9%	55.902
G_150_025	150	2835	19	2500	<b>19.127</b>	97.3%	—	—	23.718
keller4	171	5100	11	2500	<b>11.622</b>	79.3%	—	—	14.012
G_200_025	200	4905	21	2500	23.656	63.1%	—	—	28.217
brock200_1	200	5066	21	2500	22.912	70.3%	—	—	27.457
c_fat200_5	200	11427	58	2500	60.345	0.0%	—	—	60.345
sanr200_0_9	200	2037	42	2500	43.856	74.4%	—	—	49.274

Table 5.7: Results on the graphs from [97].

Graph name	$n$	$ E $	$\alpha$	SDP-Lasserre bound after		$\vartheta(G)$
				1 hour	4 hours	
rnd_p002	200	407	95	<b>95.778</b>	<b>95.244</b>	<b>95.778</b>
c_fat200_5	200	11427	58	60.345	60.317	60.345

Table 5.8: Improved SDP-Lasserre bounds with longer ADMM running times.

Graph name	$n$	$ E $	$\alpha$	$ \mathcal{B} $	SDP-Lasserre		BOUND 2 [260]		$\vartheta(G)$
					Bound	GapClsd	Bound	GapClsd	
DoubleStarSnark	30	45	13	421	<b>13.001</b>	99.9%	<b>13.001</b>	99.8%	<b>13.735</b>
Wells	32	80	10	449	<b>10.001</b>	99.9%	<b>10.906</b>	54.7%	<b>12.000</b>
Sylvester	36	90	12	577	<b>12.001</b>	99.9%	<b>12.034</b>	97.7%	13.500
Szekeressnark	50	75	21	1201	<b>21.035</b>	97.7%	<b>21.359</b>	76.6%	22.537
Klein3Regular	56	84	23	1513	<b>23.059</b>	97.8%	<b>23.908</b>	67.4%	25.793
Gosset	56	756	4	841	<b>4.048</b>	97.0%	5.600	0.0%	5.600
Gritsenko	65	1040	6	1106	<b>6.097</b>	95.2%	8.062	0.0%	8.062
Meredith	70	140	34	2346	<b>34.094</b>	80.7%	<b>34.003</b>	99.3%	<b>34.489</b>
BrouwerHaemers	81	810	15	2500	<b>15.041</b>	99.3%	20.259	12.3%	21.000
HigmanSims	100	1100	22	2500	<b>22.005</b>	99.8%	26.667	0.0%	26.667
BiggsSmith	102	153	43	2500	<b>43.536</b>	85.4%	44.270	65.5%	46.686
Balaban11Cage	112	168	52	2500	<b>52.124</b>	92.2%	<b>52.064</b>	95.9%	53.595

Table 5.9: Results on several graphs from the SageMath software.

## 5.6 Conclusions

We have considered SDP bounds on the stability number of graphs obtained from the Lasserre hierarchy at relaxation level 2, or relaxation levels intermediate to levels 1 and 2. Most of the SDPs considered here cannot be handled by IPMs, as they require excessive memory due to the large number of constraints. Therefore, we compute the bounds using the ADMM. The main operations of the ADMM algorithm are the projection onto the PSD cone (5.14), and the projection onto half-spaces, see Lemma 5.2. Although the former projection is significantly more computationally expensive than the latter, it is still manageable for the problem sizes considered in this chapter. For improved performance, our ADMM algorithm also uses warm-starting.

For Lasserre levels intermediate to levels 1 and 2, we propose a method to select a basis of variables for the relaxation, see Section 5.4.1. We use this basis selection method to choose bases of size at most 2500 for computing bounds on  $\alpha(G)$  for various graphs in Section 5.5.3. With this basis size, the ADMM algorithm often converges within one hour of computation time.

The computational experiments in Section 5.5.3 show that the Lasserre hierarchy bounds, computed via the ADMM and referred to as SDP-Lasserre bounds, are competitive with other SDP-based stable set approaches from the literature, specifically [98] and [260]. In particular, our approach provides the strongest known SDP bounds on  $\alpha(G)$  for a variety of graphs, see Tables 5.4 to 5.9. For some large and dense graphs in Table 5.7, the bound from [260] cannot be computed using the IPM due to its large memory requirement, in contrast to our SDP-Lasserre bounds.

As future work, it would be interesting to evaluate bounds from the intermediate level Lasserre hierarchy on the stability number of highly symmetric graphs. Specifically, our basis selection method from Section 5.4.1 is currently not suited to exploit symmetry in the underlying graph. Preliminary computational experiments have shown that the current basis selection method can be significantly improved for highly symmetric graphs. Another future research direction is to investigate faster methods for projecting onto the PSD cone, which is the bottleneck of the ADMM. If we find an algorithm for projecting onto the PSD cone that is better suited for the ADMM than the standard eigendecomposition, the ADMM could speed up significantly.

## 6 On solving the MAX-SAT problem using sum of squares

In this chapter, we investigate semidefinite programming (SDP) approaches for the satisfiability (SAT) and maximum-satisfiability (MAX-SAT) problems, and their variants. Given a set of logical clauses, the SAT problem is to decide whether there exists a truth assignment to the variables such that all clauses are satisfied. The optimization variant of the SAT problem, known as the MAX-SAT problem, is to determine a truth assignment which satisfies the largest number of clauses.

The SAT problem is a central problem in mathematical logic and computer science and finds various applications, including software or hardware verification [216] and planning in artificial intelligence [160]. The SAT problem was the first problem shown to be NP-complete in 1971 [60]. Since the SAT problem is NP-complete, any problem contained in the complexity class NP can be efficiently recast as a SAT instance. Thus, algorithms for the SAT problem can also solve a wide variety of other problems, such as timetabling [19, 109] and product line engineering [225].

SDP approaches to the SAT problem were first proposed by de Klerk et al. [68], and later extended by Anjos [9, 10, 11, 12]. Goemans and Williamson [117] were first to apply SDP to the MAX-SAT problem. They showed that for a specific class of MAX-SAT instances, known as MAX-2-SAT (in the MAX- $k$ -SAT problem, each clause is a disjunction of at most  $k$  literals), the MAX-SAT problem is equivalent to optimizing a multivariate quadratic polynomial, which is naturally well suited for semidefinite relaxations. In the same paper, Goemans and Williamson proposed a 0.878-approximation algorithm for the MAX-2-SAT problem based on SDP. This result was later improved to 0.940 in [192]. Further, Karloff and Zwick [157] obtained an optimal  $7/8$  approximation algorithm for the MAX-3-SAT problem. Halperin and Zwick [134] obtained a nearly optimal approximation algorithm for the MAX-4-SAT problem. In [299], van Maaren et al. exploit sum of squares (SOS) optimization to compute bounds for the MAX-SAT problem.

Despite the great success in designing approximation algorithms using SDP, most modern MAX-SAT solvers do not exploit SDP. A possible reason for this is the fact that medium to large size SDP problems (SDPs) are computationally challenging to solve. Interior-point methods, the conventional approach for solving SDPs, struggle from large memory requirements and prohibitive computation time per iteration already for medium size SDPs. Recently, first-order methods such as the alternating direction method of multipliers (ADMM) [34, 100] and the Peaceman-Rachford splitting

method (PRSM) [251] showed a great success in solving SDPs, see e.g., [69, 121, 243]. Motivated by those results, we design a MAX-SAT solver that incorporates SDP bounds and the PRSM within a branch & bound (B&B) scheme.

In particular, we further exploit the SOS approach from [299] to derive SOS-based SDP relaxations that provide strong upper bounds to the optimal MAX-SAT solution. The derived SDP relaxations are strengthened SDP duals of the Goemans and Williamson MAX-SAT relaxation. The strength of the upper bounds and the required time to compute the relaxations depend on the chosen monomial basis. We experiment with different monomial bases and propose a class of bases that provide good trade-offs between these effects. Moreover, we derive several properties of monomial bases that are exploited in the design of our solver. We extend the SOS approach to the weighted partial MAX-SAT problem, a variant of the MAX-SAT problem in which clauses are divided in soft and hard clauses. Here, the goal is to maximize the weighted sum of the satisfied soft clauses, over truth assignments that satisfy all the hard clauses. We strengthen SDP bounds for the weighted partial MAX-SAT problem using the SAT resolution rule. To the best of our knowledge, we are the first to exploit SDP for solving the weighted partial MAX-SAT problem.

We show that the PRSM is well suited for exploiting the structure of the SOS-based SDP relaxations. Therefore, we implement the PRSM to (approximately) solve large-scale SDP relaxations and obtain upper bounds for the (weighted partial) MAX-SAT problem. The resulting algorithm is very efficient, e.g., it can compute upper bounds with matrix variables of order 1800 in less than 2 minutes, and for matrices of order 2400 in less than 4 minutes. Our numerical results show that the upper bounds are strong, in particular when larger monomial bases is used. We also exploit the output of the PRSM to efficiently compute lower bounds for the MAX-SAT problem.

We design an SOS-SDP based MAX-SAT solver (named SOS-MS) that exploits SOS-based SDP relaxations and the PRSM. SOS-MS is one of the first SDP-based MAX-SAT solvers. The only alternative SDP-based solver is the MIXSAT algorithm [310] that is designed to solve MAX-2-SAT instances. SOS-MS is able to solve (weighted partial) MAX- $k$ -SAT instances, for  $k \leq 3$ . To solve a MAX-SAT instance, SOS-MS has to approximately solve multiple SDP subproblems. A crucial component of SOS-MS is therefore its ability to quickly construct the program parameters of the required SDPs, i.e., the process of *parsing*. We design an efficient parsing method, which is also applicable to other problems and publicly available. Another efficient feature of our solver is warm starts. Namely, our solver uses the approximate PRSM solution at a node, as warm starts for the corresponding children's node. We are able to solve a variety of MAX-SAT instances in a reasonable time, while solving some instances faster than the best solvers in the Eleventh Evaluation of MAX-SAT solvers (MSE-2016). Moreover, we solve three previously unsolved MAX-3-SAT instances from the MSE-2016. Our results provide new perspectives on solving the MAX-SAT problem, and all its variants, by using SDPs.

This chapter also provides various theoretical results. We propose a family of semidefinite feasibility problems, and show that one member of this family provides the rank two guarantee. That is, whenever the semidefinite relaxation admits a feasible matrix of rank two or less, the underlying SAT instance is satisfiable. This result

relates to a similar rank two guarantee result by Anjos [9]. The rank value can be seen as a measure of the strength of the relaxation. We also provide a parametric family of semidefinite relaxations for the (weighted partial) MAX-SAT problem. The parameter can be finely tuned to adjust the strength of the relaxation, and any such relaxation can easily be incorporated within SOS-MS. This allows the solver to be adapted per (class of) problem instance(s).

Further, we show how the SOS approach to the MAX-SAT problem of van Maaren et al. [299] and the, here generalized, moment relaxations of the SAT problem due to Anjos [9, 10, 11, 12] are related. This is done by exploiting the duality theory of the moment and SOS approaches. Our result generalizes a result by van Maaren et al. [299], who showed the connection between the two approaches only for restricted cases. By exploiting duality theory, we also relate the SOS relaxations for the partial MAX-SAT problem to the SAT relaxations from [9].

Lastly, we investigate MAX-SAT resolution, a powerful technique used by many MAX-SAT solvers [2], in relation to the SDP approach to the MAX-SAT problem. Standard MAX-SAT solvers use resolution to determine upper bounds on the MAX-SAT solution, while SOS-MS determines upper bounds through SDP. We show how resolution is related to the monomial basis. We also show how the SAT resolution can be exploited for the weighted partial MAX-SAT problem.

This chapter is organized as follows. We provide preliminaries in Section 6.1 and assumptions in Section 6.1.1. Section 6.2 provides an overview of the Goemans and Williamson approach [117] to the MAX-SAT problem. Section 6.3 first outlines previous SDP approaches to the SAT problem, and then generalizes them. Section 6.4 provides the details of the SOS theory, applied to the MAX-SAT problem. We also derive various properties of monomial bases in that section. Section 6.5 concerns the combination of MAX-SAT resolution and SOS. In Section 6.6, we show how two SDP approaches to SAT and MAX-SAT problems, i.e., [9] and [299], are connected. Section 6.7 introduces the PRSM for SOS. We extend the SOS approach to the weighted partial MAX-SAT problem and connect the resulting program to the relaxations in [9], in Section 6.8. Section 6.9 provides an overview and pseudocode of our solver SOS-MS. Section 6.10 presents Numerical results that include SOS-SDP bounds and performance of SOS-MS. Concluding remarks are given in Section 6.11.

## 6.1 Preliminaries

We denote by  $\phi$  a propositional formula, in variables  $x_1$  up to  $x_n$ ,  $n \in \mathbb{N}$ , and assume that  $\phi$  is in *conjunctive normal form* (CNF). That is,  $\phi$  is given by a conjunction of  $m$  clauses,

$$\phi = \bigwedge_{j=1}^m C_j, \quad (6.1)$$

where  $\wedge$  denotes the logical *and*. We will mostly use  $n$  to refer to the number of variables and  $m$  to refer to the number of clauses. Each clause  $C_j$  is a disjunction of (possibly negated) variables. We define each clause  $C_j$  as a subset of  $[n]$ , indicating

the variables appearing in  $C_j$ . Moreover, we define  $I_j^+ \subseteq C_j$  as the set of unnegated variables appearing in  $C_j$ . Similarly,  $I_j^- \subseteq C_j$  is defined as the set of negated variables appearing in  $C_j$ . Thus, the clause associated with  $C_j$  reads

$$\bigvee_{i \in I_j^+} x_i \vee \bigvee_{i \in I_j^-} \neg x_i, \quad (6.2)$$

where  $\vee$  and  $\neg$  denote the logical *or* and *negation* respectively. We refer to both  $x_i$  and  $\neg x_i$  as literals. For example, the literal  $\neg x_i$  is true if  $x_i$  is **false**. We denote the length of a clause by  $\ell_j$ , thus  $\ell_j := |C_j|$ . We say that  $\phi$  constitutes a (MAX-)k-SAT instance if  $\max_{j \in [m]} \ell_j = k$ . We associate to each clause a vector  $a_j \in \{0, \pm 1\}^n$ , having entries  $a_{j,i}$ ,  $i \in \mathbb{N}$ , according to

$$a_{j,i} = \begin{cases} -1, & \text{if } i \in I_j^-, \\ 0, & \text{if } i \notin I_j^+ \cup I_j^-, \\ 1, & \text{if } i \in I_j^+. \end{cases} \quad (6.3)$$

The SAT problem is to decide, given  $\phi$ , whether a satisfying truth assignment to the variables  $x_i$ ,  $i \in [n]$  exists. The MAX-SAT problem is to find an assignment which satisfies the largest number of clauses.

### 6.1.1 Assumptions on logical propositions

In the rest of this chapter, we assume that all logical propositions  $\phi$ , on  $n$  variables and  $m$  clauses, satisfy the following three properties:

1.  $I_j^+ \cap I_j^- = \emptyset$ ,  $\forall j \in [m]$ ,
2.  $|C_j| \geq 2$ ,  $\forall j \in [m]$ ,
3. Each variable is contained in at least 2 clauses,

along with  $\phi$  being in CNF. We explain now that properties 1 and 3 can be assumed without loss of generality. Property 1 states that a clause cannot contain both the negated and unnegated variants of a variable. Note that clauses that contain both the negated and unnegated variants of a variable are trivially satisfied by any truth assignment. For property 3, if we have that a variable occurs in exactly one clause, say  $C_j$ , we can set that variable to the truth value such that  $C_j$  is satisfied and remove  $C_j$  from  $\phi$ .

Property 2 can be assumed for SAT instances. If a SAT instance contains a clause  $C_j$  with  $|C_j| = 1$  (such a clause is known as a *unit clause*), the literal in  $C_j$  must be satisfied in any satisfying assignment. The variable corresponding to this literal can thus be given the appropriate truth value and  $\phi$  can be reduced (such a reduction of  $\phi$  is referred to as *unit resolution*). For MAX-SAT instances, it is possible that an optimal truth assignment might leave unit clauses unsatisfied. We note however, that the MAX-SAT benchmark instances we consider satisfy the properties 1 to 3.

## 6.2 MAX-SAT formulations and relaxations

We outline the approach of Goemans-Williamson for formulating the MAX-SAT problem as a polynomial optimization problem. We also present their SDP relaxation for the MAX-2-SAT problem.

Let  $x_1, x_2, \dots, x_n$  be the variables of the MAX-SAT instance, given by a logical proposition  $\phi$ . We thus assume that  $\phi$  is given in conjunctive normal form, see (6.1), and contains  $m$  clauses. As customary in the SDP SAT literature, we associate  $+1$  with **true** and  $-1$  with **false**. An assignment of the  $x_i$  values in  $\{\pm 1\}$  is referred to as a truth assignment. As proposed by Goemans and Williamson [117], we define a truth function  $v : \{\pm 1\}^n \rightarrow \{0, 1\}$ , such that, given a logical proposition  $\phi'$ , evaluated for some truth assignment,  $v(\phi') = 1$  if and only if  $\phi'$  is satisfied, and 0 otherwise. This property uniquely determines  $v$ , i.e.,

$$v(x_i) = \frac{1 + x_i}{2} \text{ and } v(\neg x_i) = \frac{1 - x_i}{2}.$$

And in general, for a clause  $C_j \subseteq [n]$  of length  $\ell_j$ , we have

$$v(C_j) := 1 - \prod_{i \in I_j^+} v(\neg x_i) \prod_{i \in I_j^-} v(x_i) = 1 - \frac{1}{2^{\ell_j}} \left( \sum_{\gamma \subseteq C_j} (-1)^{|\gamma|} a_j^\gamma x^\gamma \right), \quad (6.4)$$

for  $a_j$  as in (6.3),

$$x^\gamma := \prod_{i \in \gamma} x_i, \quad (6.5)$$

and  $a_j^\gamma$  defined similarly to (6.5). The last equality in (6.4) follows from the product expansion of  $v(C_j)$ , as shown in [11, Prop. 1]. In [117], an extra variable  $x_0 \in \{\pm 1\}$  is defined, with the purpose of deciding the truth value, that is,  $\phi'$  is true if and only if  $v(\phi') = x_0$ . We set  $x_0 = 1$  without loss of generality for sake of clarity. The MAX-SAT problem, induced by  $\phi$ , is to maximize the following polynomial:

$$v_\phi := \sum_{j \in [m]} v(C_j) = \sum_{\alpha \subseteq [n]} v_\phi^\alpha x^\alpha, \quad (6.6)$$

subject to  $x_i \in \{\pm 1\}$  for all  $i$ , and for appropriate  $v_\phi^\alpha \in \mathbb{R}$ , and  $x^\alpha$  as in (6.5). Observe that  $v_\phi$  is a  $k$ th degree polynomial if  $\phi$  represents a MAX- $k$ -SAT instance. A MAX-2-SAT instance thus corresponds to a quadratic polynomial, and is therefore well suited for SDP relaxations. We return to  $v_\phi$  in Section 6.6.

Assuming now that  $\phi$  represents a MAX-2-SAT instance on  $n$  variables, the corresponding MAX-2-SAT problem can be formulated as

$$\max \langle W, X \rangle \text{ s.t. } \text{diag}(X) = \mathbf{1}, X \succeq 0, X \in \{\pm 1\}^{(n+1) \times (n+1)}, \quad (6.7)$$

where  $W \in \mathcal{S}^{n+1}$  is the fixed matrix such that  $\langle W, \begin{bmatrix} 1 \\ x \end{bmatrix} \begin{bmatrix} 1 \\ x \end{bmatrix}^\top \rangle = v_\phi$ . Any  $X$  feasible to (6.7) satisfies  $X = xx^\top$ , for some  $x \in \{\pm 1\}^{n+1}$  [8, Thm. 2.1.1]. The size of this

vector  $x$  is one more than the number of variables  $n$ , to account for the additional truth value variable  $x_0$ .

A semidefinite relaxation of (6.7) is obtained by omitting the integrality constraint, or equivalently nonconvex rank one constraint. This constitutes the well-known Goemans-Williamson [117] SDP relaxation of the MAX-2-SAT problem. That is,

$$\max \langle W, X \rangle \text{ s.t. } \text{diag}(X) = \mathbf{1}, X \in \mathcal{S}_+^{n+1}. \quad (6.8)$$

Goemans and Williamson showed that the optimal matrix for (6.8) can be used to obtain a 0.878-approximation algorithm to the MAX-2-SAT problem. Assuming  $P \neq NP$ , for any  $\varepsilon > 0$  there exists no  $(\frac{21}{22} + \varepsilon)$  approximation algorithm to the MAX-2-SAT problem [135]. Karloff and Zwick [157] introduce a canonical way of obtaining SDP relaxations for any MAX-SAT problem, that is exploited to obtain approximation algorithms to the MAX-3-SAT and MAX-4-SAT problems in [157] and [134] respectively. To solve MAX-2-SAT problems, rather than approximate, Wang and Zico Kolter [310] propose the MIXSAT algorithm, which combines (6.8) with a B&B scheme.

### 6.3 The SAT problem as a semidefinite feasibility problem

In this section we first present a brief overview of the work done by de Klerk et al. [68] and Anjos [9, 10, 11, 12]. These works present relaxations of the SAT problem, that involve semidefinite feasibility problems. Infeasibility of their SDP relaxations implies unsatisfiability of the corresponding SAT instance. The differences between the proposed relaxations is the size of the SDP variable, and the method of encoding the structure of the SAT instance in the SDP relaxation. We propose a family of semidefinite feasibility problems, that contains relaxations from [9, 10, 11, 12, 68] as special cases, and show that a particular member of the family provides a rank-two guarantee, see Theorem 6.1.

We reconsider first program (6.8), which attempts to satisfy the maximum number of clauses through its objective function. For the SAT problem specifically, one can move the clause satisfaction part from the objective to the feasible set of a semidefinite program. This idea was first proposed by de Klerk et al. [68] in 2000, and was later extended by Anjos [9]. To be precise: de Klerk et al. propose the so called GAP relaxation, or GAP for short, which is a semidefinite feasibility problem, given by

$$\begin{aligned} \text{find } & Y \in \mathcal{S}_+^n, y \in \mathbb{R}^n \\ \text{s.t. } & a_j^\top Y a_j - 2a_j^\top y \leq \ell_j(\ell_j - 2) \quad \forall j \in [m] \\ & \text{diag}(Y) = \mathbf{1}_n \\ & Y \succeq yy^\top, \end{aligned} \quad (\text{GAP})$$

for  $a_j$  as in (6.3). It is noted in [68], that for  $\ell_j \leq 2$ , the corresponding inequalities in GAP relaxation may be changed to equalities. The GAP relaxation is suited for

instances that contain a clause of length two. If  $\ell_j \geq 3, \forall j \in [m]$ , then  $(Y, y) = (\mathbf{I}, \mathbf{0})$  is always feasible for GAP, whether the underlying SAT instance is satisfiable or not.

We now state the SDP relaxations of the SAT problem by Anjos [9, 10, 11, 12] that are not restricted to the lengths of the clauses in instances. Let  $\phi$  be a proposition on  $n$  variables and  $m$  clauses and  $x \in \{\pm 1\}^n$  the truth assignment to the variables. Consider a family of subsets  $\mathcal{B} = \{\alpha_1, \dots, \alpha_s\}, \alpha_i \subseteq [n]$ , let  $\mathbf{x} = (x^{\alpha_1}, \dots, x^{\alpha_s})^\top$ , and define  $Y := \mathbf{x}\mathbf{x}^\top$ . It is clear that  $\text{rk}(Y) = 1, \text{diag}(Y) = \mathbf{1}$ , and  $Y \succeq 0$ . Later, to obtain a semidefinite relaxation of the SAT problem, we omit the rank one constraint.

We index the matrix  $Y$  with the elements of  $\mathcal{B}$ , and define for all subsets  $\gamma$  contained in some clause of  $\phi$ , the expression

$$Y(\gamma) := Y_{\alpha, \beta}, \text{ for some } \alpha, \beta \in \mathcal{B} \text{ jointly contained in a single clause,} \tag{6.9}$$

$$\text{such that } \alpha \Delta \beta = \gamma,$$

where  $\Delta$  is the symmetric difference operator. The symmetric difference operator is induced by the fact that, for  $x \in \{\pm 1\}^n$ , we have  $Y_{\alpha, \beta} = x^\alpha x^\beta = x^{\alpha \Delta \beta} = x^\gamma$ , see (6.5). In general,  $Y(\emptyset)$  refers to a diagonal entry of  $Y$ , hence,  $Y(\emptyset) = 1$ . We may have  $Y(\gamma) = Y_{\emptyset, \gamma}$ , and we assume that, for all  $\gamma$  contained in a clause of  $\phi$ , we can always find  $\alpha$  and  $\beta$  as in (6.9).

The expression  $Y(\gamma)$  can refer to multiple entries of  $Y$ . By construction of  $Y$ , these entries are equal. Stated formally, we have  $Y \in \cap_{j \in [m]} \Delta_j$ , where

$$\Delta_j := \left\{ Y \in \mathcal{S} : \begin{array}{l} Y_{\alpha_1, \beta_1} = Y_{\alpha_2, \beta_2} \quad \forall (\alpha_1, \alpha_2, \beta_1, \beta_2) \in \mathcal{B} \\ \text{such that } \alpha_1 \Delta \beta_1 = \alpha_2 \Delta \beta_2 \subseteq C_j \end{array} \right\}. \tag{6.10}$$

Observe that the sets  $\Delta_j$  do not capture all equalities present in  $Y$ , due to the restriction  $\alpha_1 \Delta \beta_1 = \alpha_2 \Delta \beta_2 \subseteq C_j$ . In this section, we choose to include only the equalities captured by  $\Delta_j$ . This keeps the relaxations in line with previous relaxations by Anjos [9], and these equalities suffice to prove the main theorem in this section, see Theorem 6.1. In Section 6.6, we consider an SDP relaxation of the SAT problem which considers all equalities present in  $Y$ .

If  $x$  is a satisfying assignment to  $\phi$ , then  $v(C_j) = 1$ , see (6.4), for all  $j \in [m]$ . We can rewrite this constraint in terms of  $Y(\gamma)$ , see (6.9). We now omit the rank one constraint on  $Y$ , to obtain the following semidefinite feasibility program, denoted  $R_{\mathcal{B}}(\phi)$ :

$$\begin{array}{ll} \text{find} & Y \in \mathcal{S}_+^{|\mathcal{B}|} \\ \text{s.t.} & \sum_{\gamma \subseteq C_j} (-1)^{|\gamma|} a_j^\gamma Y(\gamma) = 0 \quad \forall j \in [m] \\ & \text{diag}(Y) = \mathbf{1}, Y \in \bigcap_{j \in [m]} \Delta_j. \end{array} \tag{R_{\mathcal{B}}(\phi)}$$

The program  $R_{\mathcal{B}}(\phi)$  contains both the GAP relaxation, and the relaxations proposed by Anjos [9] as special cases. Specifically, one obtains the GAP relaxation from  $R_{\mathcal{B}}(\phi)$  by taking  $\mathcal{B} = \{\alpha \subseteq [n] : |\alpha| \leq 1\}$ . For 2-SAT instances, GAP is feasible if and only if the corresponding 2-SAT instance is satisfiable [68]. Note that 2-SAT is decidable in linear time [20], unlike the NP-complete  $k$ -SAT,  $k \geq 3$ .

The GAP relaxation can be considered as a semidefinite program in the first level of the well-known Lasserre hierarchy [175]. Anjos [9] proposed semidefinite relaxations of the SAT problem in approximately levels two and three of the Lasserre hierarchy, by only adding a subset of products of variables to the moment relaxation. For example, Anjos [9] proposed the  $R_2$  relaxation, which can be obtained from  $R_{\mathcal{B}}(\phi)$  by taking

$$\mathcal{B} = \{\alpha : \alpha \subseteq C_j \text{ for some } j, |\alpha| \text{ odd, or } \alpha = \emptyset\}. \quad (6.11)$$

It was proved in [9] that the  $R_2$  relaxation attains a rank two guarantee on 3-SAT instances: whenever the SDP admits a feasible matrix of rank two or lower, the corresponding 3-SAT instance is satisfiable. We will now prove that, for a different  $\mathcal{B}$  than (6.11), the resulting relaxation  $R_{\mathcal{B}}(\phi)$  provides the same rank two guarantee.

**Theorem 6.1.** *Let  $\phi$  be a 3-SAT instance and*

$$\mathcal{B} = \{\alpha \subseteq [n] : \alpha \subseteq C_j \text{ for some } j, |\alpha| \leq 2\}. \quad (6.12)$$

*If the SDP relaxation  $R_{\mathcal{B}}(\phi)$  admits a feasible rank two matrix, then  $\phi$  is satisfiable.*

*Proof.* The proof is adapted from [9, Thm. 3], in which the theorem is proven for the case that  $\mathcal{B}$  is given as in (6.11).

Since  $\phi$  is a 3-SAT instance, there exists a clause of length three. Fix a  $j$  for which clause  $C_j = \{i_1, i_2, i_3\}$  and set

$$\mathcal{B}_j := \{\alpha \subseteq [n] : \alpha \subseteq C_j, |\alpha| \leq 2\} = \{\{\emptyset\}, \{i_k\}_{k \in \{1,2,3\}}, \{i_1, i_2\}, \{i_1, i_3\}, \{i_2, i_3\}\},$$

for  $\mathcal{B}$  as in (6.12). Note that  $\mathcal{B}_j \subseteq \mathcal{B}$ . Let  $Y$  be feasible solution to  $R_{\mathcal{B}}(\phi)$  of rank 2. Consider the submatrix of  $Y$ , indexed by some of the elements of  $\mathcal{B}_j$ ,

$$\begin{bmatrix} 1 & Y(i_1 i_2) & Y(i_1 i_3) & Y(i_2 i_3) \\ Y(i_1 i_2) & 1 & Y(i_2 i_3) & Y(i_1 i_3) \\ Y(i_1 i_3) & Y(i_2 i_3) & 1 & Y(i_1 i_2) \\ Y(i_2 i_3) & Y(i_1 i_3) & Y(i_1 i_2) & 1 \\ Y(i_1 i_2 i_3) & Y(i_3) & Y(i_2) & Y(i_1) \end{bmatrix}, \quad (6.13)$$

where  $Y(\cdot)$  is as in (6.9). For example,  $Y(i_1 i_2) = Y_{\emptyset, i_1 i_2}$  and  $Y(i_1) = Y_{i_2 i_3, i_1 i_2 i_3}$ . As the matrix in (6.13) has rank at most 2, it can be proven [14, Lem. 3.11] that at least one of  $Y(i_1 i_2)$ ,  $Y(i_1 i_3)$ ,  $Y(i_2 i_3)$  equals  $\delta$ , where  $\delta \in \{\pm 1\}$ .

Assume without loss of generality that  $Y(i_1 i_2) = \delta$ . Consider now the following  $3 \times 3$  principal submatrices of  $Y$ :

$$\begin{bmatrix} 1 & Y(i_3) & Y(i_1 i_2) \\ & 1 & Y(i_1 i_2 i_3) \\ & & 1 \end{bmatrix}, \begin{bmatrix} 1 & Y(i_1 i_2) & Y(i_2 i_3) \\ & 1 & Y(i_1 i_3) \\ & & 1 \end{bmatrix}, \begin{bmatrix} 1 & Y(i_1) & Y(i_1 i_2) \\ & 1 & Y(i_2) \\ & & 1 \end{bmatrix}, \quad (6.14)$$

where we have omitted the lower triangular part, which is fixed by symmetry. The three matrices in (6.14) have the following three properties: they are PSD, they have unit diagonal and at least one of their entries is contained in  $\{\pm 1\}$ , i.e., the entry  $Y(i_1 i_2) \in \{\pm 1\}$ . It can then be shown [14, Lem. 3.9] that

$$Y(i_1 i_2 i_3) = \delta Y(i_3), \quad Y(i_2 i_3) = \delta Y(i_1 i_3), \quad Y(i_2) = \delta Y(i_1), \quad \text{for } \delta = Y(i_1 i_2). \quad (6.15)$$

The nonlinear equalities in (6.15) allow us to simplify the satisfiability constraint on  $C_j$ , given by

$$\begin{aligned} \sum_{\gamma \subseteq C_j} (-1)^{|\gamma|} a_j^\gamma Y(\gamma) &= 1 - a_1 Y(i_1) - a_2 Y(i_2) - a_3 Y(i_3) + a_1 a_2 Y(i_1 i_2) \\ &\quad + a_1 a_3 Y(i_1 i_3) + a_2 a_3 Y(i_2 i_3) - a_1 a_2 a_3 Y(i_1 i_2 i_3) \\ &= 0, \end{aligned} \tag{6.16}$$

see the constraints of  $R_{\mathcal{B}}(\phi)$ . Here, we have written  $a_k$  for  $a_{j,i_k}$ ,  $k \in \{1, 2, 3\}$ , see (6.3). Substituting (6.15) in (6.16) yields

$$\left[1 + a_1 a_2 \delta\right] \left[1 - a_1 Y(i_1) - a_3 Y(i_3) + a_1 a_3 Y(i_1 i_3)\right] = 0, \tag{6.17}$$

where we have used that  $a_k^2 = 1$ . Note that  $1 + a_1 a_2 \delta \in \{0, 2\}$ , hence, there are two cases to consider. In the case that  $1 + a_1 a_2 \delta = 2$ , equation (6.17) reduces to

$$1 - a_1 Y(i_1) - a_3 Y(i_3) - a_2 a_3 Y(i_2 i_3) = 0, \tag{6.18}$$

which is a linear constraint in the entries of matrix  $Y$ . In case  $1 + a_1 a_2 \delta = 0$ , clause  $C_j$  is satisfied when  $x_{i_1} x_{i_2} = Y(i_1 i_2) = \delta$ .

We have shown in (6.17) that for any clause  $C_j$  of length three, its corresponding linear satisfiability constraint (6.16), can be written as  $g_j(\delta) f_j(Y) = 0$  when  $\text{rk}(Y) = 2$ , for polynomials  $g_j$  and  $f_j$ . For all  $j$  such that  $g_j(\delta) \neq 0$ , let  $B_j$  be the set of the singletons  $\alpha \in \mathcal{B}$ ,  $|\alpha| \leq 1$ , that also appear as  $Y(\alpha)$  in the polynomial  $f_j(Y)$ . Note that for the particular equation (6.17), we would have  $B_j = \{\{i_1\}, \{i_3\}\}$ . Note also that (6.18) is the constraint of the semidefinite relaxation of a clause of length two on the variables  $x_{i_1}$  and  $x_{i_3}$ .

Let  $B$  be the union of all  $B_j$  and consider the submatrix of  $Y$  indexed by all sets in  $B$  and the set  $\emptyset$ . As  $Y$  is feasible for  $R_{\mathcal{B}}(\phi)$ ,  $Y$  is automatically feasible for the GAP relaxation corresponding to some 2-SAT instance on the variables  $x_i$ ,  $i \in B$ , which implies satisfiability of the corresponding 2-SAT instance [68, Thm. 5.1]. This implies that the  $x_i$  variables have a truth assignment that satisfy the reduced clauses of length two. This truth assignment to the  $x_i$ ,  $i \in B$ , variables can be extended to a truth assignment to the variables  $x_i$ ,  $i \notin B$ , by using the appropriate values of  $\delta$  and (6.15). This proves the theorem.  $\square$

We will return to the basis (6.12) in Section 6.4.1, where we study it for the purpose of solving MAX-SAT instances.

## 6.4 Sum of squares and the MAX-SAT problem

In Section 6.4.1 we first provide an overview of the approach of van Maaren et al. [299] for deriving relaxations for the MAX-SAT problem. Their approach exploits SOS optimization, which has received much attention in the literature, see e.g., [176, 185, 250, 270]. Relaxations depend on a basis of monomials that is used to compute them.

We introduce a parametric family of monomial bases with increasing complexity. In Section 6.4.2 we derive several properties of monomial bases that are later used in our computations.

### 6.4.1 General overview

For a given logical proposition  $\phi$ , on  $n$  variables and  $m$  clauses, the value

$$F_\phi(x) := \sum_{j=1}^m \frac{1}{2^{\ell_j}} \prod_{i \in C_j} (1 - a_{j,i} x_i), \quad (6.19)$$

for  $a_{j,i}$  as in (6.3), equals the number of unsatisfied clauses by truth assignment  $x \in \{\pm 1\}^n$ . Hence, we are interested in minimizing  $F_\phi$  over  $\{\pm 1\}^n$ , on which  $F_\phi$  is nonnegative. Let  $\mathbb{R}[x]$  be the set of real polynomials in  $x_1, \dots, x_n$ . We define

$$\mathcal{V} := \left\{ f : f \equiv \sum_{j=1}^k f_j^2 \pmod{\mathcal{I}}, f_j \in \mathbb{R}[x] \forall j \in [k], k \in \mathbb{N} \right\} \quad (6.20)$$

as the set of SOS polynomials modulo  $\mathcal{I}$ , where  $\mathcal{I}$  is the vanishing ideal of  $\{\pm 1\}^n$ . That is

$$\mathcal{I} = \langle 1 - x_1^2, 1 - x_2^2, \dots, 1 - x_n^2 \rangle. \quad (6.21)$$

By *Putinar's Positivstellensatz* [261],  $\mathcal{V}$  is the set of nonnegative polynomials on  $\{\pm 1\}^n$ . Generally, optimization over  $\mathcal{V}$  is intractable due to its size, which is why we consider

$$\mathcal{V}_{\mathbf{x}} := \{ f : f \equiv \mathbf{x}^\top M \mathbf{x} \pmod{\mathcal{I}}, M \succeq 0 \}, \quad (6.22)$$

where  $\mathbf{x}$  is some monomial basis. Since  $M \succeq 0$ , it follows that all polynomials of  $\mathcal{V}_{\mathbf{x}}$  are nonnegative on  $\{\pm 1\}^n$ . Therefore,  $\mathcal{V}_{\mathbf{x}} \subseteq \mathcal{V}$ , and we may approximate the minimum of  $F_\phi$  by

$$\min_{x \in \{\pm 1\}^n} F_\phi = \sup \{ \mu \in \mathbb{R} : F_\phi - \mu \in \mathcal{V} \} \geq \sup \{ \mu \in \mathbb{R} : F_\phi - \mu \in \mathcal{V}_{\mathbf{x}} \}. \quad (6.23)$$

The description of  $\mathcal{V}_{\mathbf{x}}$  shows that the lower bound in (6.23) can be computed via SDP.

It is important to note that in the quotient ring of  $\mathbb{R}[x]$  modulo  $\mathcal{I}$ , all terms  $x_i^2 \equiv 1$ , and thus it suffices to consider only monomials in  $\mathbf{x}$  for which the largest power is at most 1. Thus, we can write

$$F_\phi(x) = \sum_{\alpha \subseteq [n]} p_\phi^\alpha x^\alpha, \quad (6.24)$$

where  $p_\phi^\alpha \in \mathbb{R}$  for all  $\alpha \subseteq [n]$  and  $x^\alpha$  as in (6.5). For the constant term of  $F_\phi(x)$ , we have

$$p_\phi^\emptyset = \sum_{j=1}^m \frac{1}{2^{\ell_j}}. \quad (6.25)$$

We say that monomial basis  $\mathbf{x}$  represents a logical proposition  $\phi$  if matrix  $\mathbf{X} \equiv \mathbf{x}\mathbf{x}^\top \pmod{\mathcal{I}}$  contains all monomials  $x^\alpha$  for which  $p_\phi^\alpha \neq 0$ . We index this matrix  $\mathbf{X}$  and the matrix  $M$  from (6.22) with subsets  $\alpha \subseteq [n]$  for which  $x^\alpha \in \mathbf{x}$ . Note that for such  $\alpha, \beta \subseteq [n]$ , we have

$$\mathbf{X}_{\alpha, \beta} \equiv x^{\alpha \Delta \beta} \pmod{\mathcal{I}}. \quad (6.26)$$

For  $\alpha \subseteq [n]$ , we write  $x^\alpha \in \mathbf{X}$  if  $\mathbf{X}$  has an entry equal to  $x^\alpha$  (modulo  $\mathcal{I}$ ). van Maaren et al. [299] propose multiple monomial bases  $\mathbf{x}$ , among them basis  $SOS_p$ , given by

$$SOS_p = 1 \cup \{x_i : i \in [n]\} \cup \{x_i x_j : i \text{ and } j \text{ jointly appear in a clause}\}. \quad (6.27)$$

It is stated in [299] that  $SOS_p$  represents 2-SAT and 3-SAT instances. While this is true, this basis also represents 4-SAT instances (see Lemma 6.3). We additionally define for  $\mathbf{Q} \in \{0\} \cup [n]$ , as extension to  $SOS_p$ , the basis

$$SOS_p^{\mathbf{Q}} := SOS_p \cup \left\{ x_i x_j : \begin{array}{l} i \text{ and } j \text{ are both in the} \\ \text{top } \mathbf{Q} \text{ appearing variables} \end{array} \right\}. \quad (6.28)$$

Basis  $SOS_p^{\mathbf{Q}}$  takes basis  $SOS_p$  and adds all the  $\binom{\mathbf{Q}}{2}$  quadratic terms of the  $\mathbf{Q}$  variables appearing in the largest number of clauses of  $\phi$ . Any basis  $\mathbf{x}$  is considered to have duplicate monomials removed, and so, for small values of  $\mathbf{Q}$ , bases  $SOS_p^{\mathbf{Q}}$  and  $SOS_p$  might coincide.

We also define the basis  $SOS_s^\theta$ , for  $\theta \in [0, 1]$ , which is suited for (MAX-)2-SAT instances. This basis consists of all the monomials of degree one and zero, plus a percentage  $\theta$  of all quadratic monomials appearing in  $SOS_p$ . The included quadratic monomials are those that appear in  $SOS_p$  and attain the largest monomial weight  $w$ , which is defined as  $w(x^\alpha) := \sum_{i \in \alpha} w(i)$ , where  $w(i) := |\{C \in \phi : i \in C\}|$ , for  $i \in [n]$ . This results in the following chain of inclusions:

$$\begin{aligned} \{x^\alpha : |\alpha| \leq 1\} &= SOS_s^0 \subseteq SOS_s^\theta \subseteq SOS_s^1 = SOS_p = SOS_p^0 \\ &\subseteq SOS_p^{\mathbf{Q}} \subseteq SOS_p^n = \{x^\alpha : |\alpha| \leq 2\}. \end{aligned} \quad (6.29)$$

We now define for all  $\gamma \subseteq [n]$  such that  $x^\gamma \in \mathbf{X}$ , a set of ordered pairs as follows

$$\mathbf{x}^\gamma := \{(\alpha, \beta) \subseteq [n]^2 : \alpha \Delta \beta = \gamma, x^\alpha \in \mathbf{x}, x^\beta \in \mathbf{x}\}. \quad (6.30)$$

Set  $\mathbf{x}^\gamma$  contains the index pairs  $(\alpha, \beta)$  such that  $\mathbf{X}_{\alpha, \beta} \equiv x^\gamma \pmod{\mathcal{I}}$ . Therefore,  $F_\phi \equiv \mathbf{x}^\top M \mathbf{x}$  if and only if

$$\sum_{(\alpha, \beta) \in \mathbf{x}^\gamma} M_{\alpha, \beta} = p_\phi^\gamma, \quad \forall \gamma \subseteq [n], \quad (6.31)$$

where we set the summation to 0 if  $\mathbf{x}^\gamma = \emptyset$ . Constraints of the form (6.31) are sometimes referred to as *coefficient matching conditions* in SOS literature [318]. We define

$$\mathcal{M}_\phi := \left\{ M \in \mathcal{S}^{|\mathbf{x}|} : \sum_{(\alpha, \beta) \in \mathbf{x}^\gamma} M_{\alpha, \beta} = p_\phi^\gamma \quad \forall \gamma \subseteq [n], \gamma \neq \emptyset \right\}, \quad (6.32)$$

as the set of matrices that satisfy the coefficient matching conditions, for all monomials except  $x^\emptyset$ .

Note that  $M$  is constrained to be symmetric, which is reflected in the definition of  $\mathbf{x}^\gamma$ , since  $(\alpha, \beta) \in \mathbf{x}^\gamma$  if and only if  $(\beta, \alpha) \in \mathbf{x}^\gamma$ . Moreover,  $\mathbf{x}^\emptyset$  contains the index pairs of the diagonal entries of  $M$ , which correspond to zero-degree monomials in  $\mathbf{X}$ . Hence,

$$F_\phi - \mu \equiv \mathbf{x}^\top M \mathbf{x} \implies M \in \mathcal{M}_\phi \text{ and } p_\phi^\emptyset - \mu = \langle \mathbf{I}, M \rangle,$$

see (6.23) and (6.25). To maximize the lower bound on  $F_\phi$ , see (6.23), we maximize  $\mu$ , which is thus equivalent to minimizing  $\langle \mathbf{I}, M \rangle$ . We can therefore compute this lower bound by solving the following SDP:

$$\min \langle \mathbf{I}, M \rangle \text{ s.t. } M \in \mathcal{M}_\phi \cap \mathcal{S}_+. \quad (\mathbf{P}_\phi)$$

We note that, for the purpose of solving  $\mathbf{P}_\phi$  through interior-point methods, program  $\mathbf{P}_\phi$  is strictly feasible: for any feasible matrix  $M$ , matrix  $M + \mathbf{I}$  is strictly feasible. The existence of any such feasible matrix  $M$  follows from the nonnegativity of  $F_\phi$  on  $\{\pm 1\}^n$ . We postpone the derivation of the dual of  $\mathbf{P}_\phi$  to Section 6.6, where we also show its strict feasibility in Theorem 6.6.

### 6.4.2 Properties of $SOS_p^{\mathbf{Q}}$

We provide several properties of monomial bases that are exploited within the PRSM, see Section 6.7.

Denote by  $|\mathbf{x}^\gamma|$  the cardinality of the set  $\mathbf{x}^\gamma$ , see (6.30). Due to the symmetry of  $\mathbf{X}$ , see (6.26),  $|\mathbf{x}^\gamma|$  is an even number and greater than or equal to 2. In particular, when  $|\mathbf{x}^\gamma| = 2$ , say  $\mathbf{x}^\gamma = \{(\alpha, \beta), (\beta, \alpha)\}$ , we have

$$M_{\alpha, \beta} + M_{\beta, \alpha} = p_\phi^\gamma \text{ and } M_{\alpha, \beta} = M_{\beta, \alpha} \implies M_{\alpha, \beta} = M_{\beta, \alpha} = p_\phi^\gamma / 2. \quad (6.33)$$

Thus, whenever  $|\mathbf{x}^\gamma| = 2$ , the constraint involving  $\mathbf{x}^\gamma$  in  $\mathcal{M}_\phi$ , see (6.32), simply fixes two entries of  $M$ . van Maaren et al. [299] refer to these constraints arising from  $|\mathbf{x}^\gamma| = 2$  as *unit constraints*. In [299, Section 7], the authors empirically show that a large percentage of the constraints of  $\mathcal{M}_\phi$  are unit constraints. The authors of [299] propose as future work the development of an SDP solver that is able to exploit the large number of unit constraints. We propose an algorithm for approximately solving  $\mathbf{P}_\phi$  in Section 6.7, which is able to do so.

The following lemma describes the subsets  $\gamma$  that induce unit constraints.

**Lemma 6.2.** *Let  $\phi$  be a (MAX-)SAT instance on  $n$  variables and  $m$  clauses, and  $\mathbf{x}$  its corresponding  $SOS_p^{\mathbf{Q}}$  basis, see (6.28), for some  $\mathbf{Q} \in \{0\} \cup [n]$ . Then, for all  $\gamma \subseteq [n]$ , we have the following implication*

$$|\mathbf{x}^\gamma| = 2 \implies p_\phi^\gamma = 0, \quad (6.34)$$

where  $\mathbf{x}^\gamma$  is as in (6.30), and  $p_\phi^\gamma$  is a coefficient of  $F_\phi(x)$ , see (6.24).

*Proof.* It follows from the definition of  $F_\phi(x)$ , see (6.19) and (6.24), that for all  $\gamma \subseteq [n]$ ,

$$\gamma \not\subseteq C_j \quad \forall j \in [m] \implies p_\phi^\gamma = 0. \quad (6.35)$$

We will prove the following implication

$$\gamma \subseteq C_j, j \in [m] \implies |\mathbf{x}^\gamma| \neq 2. \quad (6.36)$$

Then, the contrapositive of (6.36), combined with (6.35), proves (6.34).

To prove (6.36), let  $\gamma \subseteq C_j$  for some  $j \in [m]$ . We distinguish four cases based on the value of  $|\gamma|$ .

**Case 1.**  $|\gamma| \in \{0, 1, 2\}$ .

We assume without loss of generality that  $\gamma \subseteq \{i_1, i_2\} \subseteq C_j$ . Since  $\{i_1, i_2\} \subseteq C_j$ , we have that  $x^\beta \in \mathbf{x}$ , for all  $\beta \in S := \{\emptyset, i_1, i_2, \{i_1, i_2\}\}$ . We consider all the subsets obtained by taking pairwise symmetric difference of elements of  $S$  in the symmetric matrix

$$S_\Delta := \begin{bmatrix} \emptyset & i_1 & i_2 & i_1 i_2 \\ i_1 & \emptyset & i_1 i_2 & i_2 \\ i_2 & i_1 i_2 & \emptyset & i_1 \\ i_1 i_2 & i_2 & i_1 & \emptyset \end{bmatrix}.$$

Observe that for all possible  $\gamma \subseteq \{i_1, i_2\}$ , we have that  $|\mathbf{x}^\gamma| \geq 4$ , and thus,  $|\mathbf{x}^\gamma| \neq 2$ .

**Case 2.**  $|\gamma| = 3$ .

We assume without loss of generality that  $\gamma = \{i_1, i_2, i_3\} \subseteq C_j$ . By again constructing  $S_\Delta$  for  $S = \{\emptyset, i_1, i_2, i_3, \{i_1, i_2\}, \{i_1, i_3\}, \{i_2, i_3\}\}$  (details omitted), one can show that  $|\mathbf{x}^\gamma| \geq 4$ .

**Case 3.**  $|\gamma| = 4$ . Proof is similar to the case  $|\gamma| = 3$ .

**Case 4.**  $|\gamma| > 4$ . By definition of  $SOS_p^{\mathbf{Q}}$ , see (6.28),  $|\mathbf{x}^\gamma| = 0$  whenever  $|\gamma| > 4$ .  $\square$

Lemma 6.2 implies that, in an implementation which uses the  $SOS_p^{\mathbf{Q}}$  basis, it is not required to store the coefficients corresponding to unit constraints (since these all equal 0), but only the indices restricted by the unit constraints. The converse of Lemma 6.2 is generally not true. That is, there can exist many subsets  $\gamma \subseteq [n]$  for which  $p_\phi^\gamma = 0$ , but  $|\mathbf{x}^\gamma| > 2$ .

**Lemma 6.3.** *Let  $\phi$  be a (MAX-)SAT instance on  $n$  variables and  $\mathbf{x}$  its monomial basis according to  $SOS_p^{\mathbf{Q}}$ , see (6.28), for some  $\mathbf{Q} \in \{0\} \cup [n]$ . Let  $\gamma \subseteq [n]$ . Then*

1.  $|\gamma| \in \{1, 2\} \implies |\mathbf{x}^\gamma| \leq 2n$ .
2.  $|\gamma| \in \{3, 4\} \implies |\mathbf{x}^\gamma| \leq 6$ .
3.  $|\gamma| > 4 \implies |\mathbf{x}^\gamma| = 0$ .

*Proof.* Let  $\mathcal{B} := \{\beta \in [n] : x^\beta \in SOS_p^{\mathbf{Q}}\}$ . The proof follows from enumerating the pairs of sets in  $\mathcal{B}$  such that their symmetric difference equals  $\gamma$ . For Item 1 of the lemma, assume first, without loss of generality, that  $\gamma = \{1\}$ . Then consider the tuples  $(\emptyset, \{1\})$  and  $(\{1, k\}, \{k\})$ , for  $k \in [n] \setminus \{1\}$ . There are  $2n$  of these tuples (counting the symmetry of order twice) and their symmetric differences all equal  $\gamma$ . Next, we assume without loss of generality that  $\gamma = \{1, 2\}$ . Then the tuples  $(\{1\}, \{2\})$ ,  $(\{1, 2\}, \emptyset)$  and

$(\{1, k\}, \{2, k\})$ , for  $k \in [n] \setminus \{1, 2\}$ , have their pairwise symmetric difference equal to  $\gamma$ . There are  $2n$  of these tuples, which proves Item 1 of the lemma.

Assuming that  $\gamma = \{1, 2, 3\}$ , we find the 6 tuples as  $(\{i\}, \gamma \setminus \{i\})$ ,  $i \in \gamma$ . If instead  $|\gamma| = 4$ , each tuple corresponds to one of  $\binom{4}{2} = 6$  partitions which proves Item 2. Lastly, it follows from the definition of  $SOS_p^{\mathcal{Q}}$ , that any monomial in matrix  $\mathbf{xx}^T$  is of degree at most four, which proves part Item 3 of the lemma.  $\square$

Part 3 of Lemma 6.3 shows that the  $SOS_p^{\mathcal{Q}}$  bases are only suited for the (MAX-)  $k$ -SAT problem when  $k \leq 4$ .

## 6.5 Resolution and monomial bases

In this section, we consider resolution in combination with the SOS approach to the MAX-SAT problem. Resolution is a technique from mathematical logic, and widely employed by MAX-SAT solvers [259]. Resolution takes as inputs two clauses of a proposition  $\phi$ , and returns a set of new clauses, named the *resolvent* clauses. The resolvent clauses transform  $\phi$  into  $\phi'$ , by either replacing the original clauses, or by adding the resolvent clauses to  $\phi$  (depending on which resolution rule is used). We show in this section that the MAX-SAT resolution rule might not be beneficial for the SOS approach applied to the MAX-SAT problem, and can even decrease its effectiveness. However, in Section 6.8.2 we show how to benefit from the SAT resolution rule when solving partial MAX-SAT problems.

We show this using an example. For  $k \geq 3$ , we define the following proposition on  $k$  variables

$$\phi_k := \begin{cases} \neg x_1 \wedge (x_1 \vee \neg x_2) \wedge (x_2 \vee x_3) \wedge \neg x_3 & \text{if } k = 3, \\ \neg x_1 \wedge (x_1 \vee \neg x_2) \wedge (x_2 \vee x_3) \wedge \left[ \bigwedge_{j=3}^{k-1} (\neg x_j \vee x_{j+1}) \right] \wedge \neg x_k & \text{else.} \end{cases} \quad (6.37)$$

It is clear that  $\phi_k$  is unsatisfiable. If one satisfies the initial two unit clauses and performs unit resolution, more unit clauses appear. Repeating this process will lead to an all **false** truth assignment, leaving clause  $x_2 \vee x_3$  unsatisfied. Therefore, any truth assignment leaves at least one clause unsatisfied, and hence,

$$\min_{x \in \{\pm 1\}^k} F_{\phi_k} = F_{\phi_k}(-\mathbf{1}_k) = 1, \quad (6.38)$$

for  $F_{\phi_k}$  as in (6.19).

In the following lemma, we show that the  $SOS_p$  basis, see (6.27), suffices for proving optimality of this assignment.

**Lemma 6.4.** *For all  $k \geq 3$ , we have that*

$$\max \{ \mu : F_{\phi_k} - \mu \in \mathcal{V}_{\mathbf{x}} \} = 1, \quad (6.39)$$

where  $\mathbf{x} = SOS_p(\phi_k)$ , and  $\mathcal{V}_{\mathbf{x}}$  is as in (6.22).

*Proof.* Since  $\min_{x \in \{\pm 1\}^k} F_{\phi_k} = 1$ , see (6.38), there does not exist a  $\mu > 1$  such that  $F_{\phi_k} - \mu \in \mathcal{V}$ , see (6.20). As  $\mathcal{V}_{\mathbf{x}} \subseteq \mathcal{V}$ , this implies that  $\nexists \mu > 1$  such that  $F_{\phi_k} - \mu \in \mathcal{V}_{\mathbf{x}}$ . Thus, to prove (6.39) it suffices to show that  $F_{\phi_k} - 1 \in \mathcal{V}_{\mathbf{x}}$ , for all  $k \geq 3$ .

We prove this by induction. For the base case  $k = 3$ , we have

$$\begin{aligned} F_{\phi_3} &= (6 + x_1 + x_3 - x_1x_2 + x_2x_3)/4 \\ &\equiv 1 + (2 + x_1 + x_3 - x_1x_2 + x_2x_3)^2/32 \\ &\quad + (-x_1 + 2x_2 + x_3 + x_1x_2 + x_2x_3)^2/32 \pmod{\mathcal{I}}, \end{aligned} \quad (6.40)$$

for  $\mathcal{I}$  as in (6.21). Clearly, the monomials appearing in (6.40) are contained in  $SOS_p(\phi_k)$ . Therefore,  $F_{\phi_3} - 1 \in \mathcal{V}_{\mathbf{x}}$ . Now for  $k + 1$ , we have

$$\begin{aligned} F_{\phi_{k+1}} &= F_{\phi_k} + (1 - x_k + x_{k+1} - x_kx_{k+1})/4 \\ &\equiv 1 + (F_{\phi_k} - 1) + (1 - x_k + x_{k+1} - x_kx_{k+1})^2/16 \pmod{\mathcal{I}}. \end{aligned}$$

By the induction hypothesis,  $F_{\phi_k} - 1 \in \mathcal{V}_{\mathbf{x}}$ , and so it follows that  $F_{\phi_{k+1}} - 1 \in \mathcal{V}_{\mathbf{x}}$  as well.  $\square$

Let us now present the MAX-SAT resolution rule, see e.g., [2]. For clauses  $C_1$  and  $C_2$  of some proposition  $\phi$ , on literals  $x, z_i, i \in [s]$  and  $y_i, i \in [t]$  construct the clauses below the horizontal line:

$$\begin{array}{c} C_1 = [x \vee z_1 \vee \dots \vee z_s], C_2 = [\neg x \vee y_1 \vee \dots \vee y_t] \\ \hline z_1 \vee \dots \vee z_s \vee y_1 \vee \dots \vee y_t, \\ [C_1 \vee \neg y_1 \vee y_2 \vee \dots \vee y_t], [C_1 \vee \neg y_2 \vee y_3 \vee \dots \vee y_t], \dots, [C_1 \vee \neg y_t], \\ [C_2 \vee \neg z_1 \vee z_2 \vee \dots \vee z_s], [C_2 \vee \neg z_2 \vee z_3 \vee \dots \vee z_s], \dots, [C_2 \vee \neg z_s]. \end{array} \quad (6.41)$$

The MAX-SAT resolution rule states that one may replace clauses  $C_1$  and  $C_2$  in  $\phi$  with a subset of the  $1 + s + t$  resolvent clauses below the horizontal line. Namely, clauses that are trivially satisfied, such as  $x \vee \neg x$ , are not part of this subset. We refer to the resulting new proposition, obtained after resolution, as  $\phi'$ . In [32, Thm. 4], it is proven that any truth assignment leaves the same number of clauses unsatisfied for  $\phi$  and  $\phi'$ . This is referred to as *soundness* of the MAX-SAT resolution rule. By soundness and the definition of  $F_\phi$ , see (6.19), it follows that  $F_\phi = F_{\phi'}$ .

For standard MAX-SAT solvers, one of the goals of resolution is to create new unit clauses, which are used to compute upper bounds on the MAX-SAT solution [2]. For our SDP approach, assuming a fixed monomial basis, the sets  $\mathcal{M}_\phi$  and  $\mathcal{M}_{\phi'}$ , see (6.32), depend only on the coefficients of  $F_\phi$  and  $F_{\phi'}$ . Since  $F_\phi = F_{\phi'}$ , these coefficients are equal, and thus  $\mathcal{M}_\phi = \mathcal{M}_{\phi'}$ . Note that the feasible set of  $\mathbf{P}_\phi$  is defined in terms of  $\mathcal{M}_\phi$ . Hence, it follows that, if given the same basis, program  $\mathbf{P}_\phi$  equals program  $\mathbf{P}_{\phi'}$ . This equivalence of programs suggests that MAX-SAT resolution does not change our approach, however, we find that in general  $SOS_p(\phi) \neq SOS_p(\phi')$ , see (6.27). We investigate the effect of this difference.

Returning to the example of  $\phi_k$  in (6.37), let us define  $C_q = \neg x_q \vee x_{q+1}$ . Observe that for  $3 \leq q \leq k - 1$ ,  $C_q \in \phi_k$ . Let us fix some  $q$ ,  $3 \leq q \leq k - 3$ , and consider the clauses  $C_q, C_{q+1}, C_{q+2} \in \phi_k$ . We perform resolution as:

$$\begin{array}{c} C_q = [\neg x_q \vee x_{q+1}], C_{q+1} = [\neg x_{q+1} \vee x_{q+2}] \\ \hline [\neg x_q \vee x_{q+2}], [\neg x_q \vee x_{q+1} \vee \neg x_{q+2}], [x_q \vee \neg x_{q+1} \vee x_{q+2}]. \end{array} \quad (6.42)$$

We perform resolution again, on the third new clause obtained in (6.42) and  $C_{q+2}$ , to obtain:

$$\frac{[x_q \vee \neg x_{q+1} \vee x_{q+2}], C_{q+2} = [\neg x_{q+2} \vee x_{q+3}]}{[x_q \vee \neg x_{q+1} \vee x_{q+3}], [x_q \vee \neg x_{q+1} \vee x_{q+2} \vee \neg x_{q+3}],} \quad (6.43)$$

$$[\neg x_q \vee \neg x_{q+1} \vee \neg x_{q+2} \vee x_{q+3}], [x_{q+1} \vee \neg x_{q+2} \vee x_{q+3}],$$

The resolution rule states that one may replace the original clauses  $C_1$ ,  $C_2$  and  $C_3$  with the 6 new resolvent clauses obtained from (6.42) and (6.43) (the third resolvent from (6.42) is not counted, since it is replaced in the resolution in (6.43)).

Observe that the  $SOS_p$  basis generates 6 quadratic monomials for the new resolvent clauses, while originally, only 3 quadratic monomials are generated for  $C_q$ ,  $C_{q+1}$  and  $C_{q+2}$ . We now define,  $\phi'_k$  for  $k \geq 6$ , as the logical proposition, obtained by taking  $\phi_k$ , and performing resolution as in (6.42) and (6.43), for each triple of clauses  $\{C_q, C_{q+1}, C_{q+2}\}$ , for each  $q \in \{3, 6, 9, \dots, k-3\}$  (let us assume here that  $k$  is a multiple of 3). Note that proposition  $\phi'_k$  constitutes a MAX-4-SAT instance, and therefore basis  $SOS_p$  is applicable. Let us compare the sizes of the resulting  $SOS_p$  bases, denoted as  $|SOS_p|$ . We have

$$|SOS_p(\phi_k)| = 2k < 3k - 3 = |SOS_p(\phi'_k)|.$$

Thus, compared to  $SOS_p(\phi_k)$ , basis  $SOS_p(\phi'_k)$  adds approximately  $k$  monomials. None of these monomials strengthen the bound, since  $SOS_p(\phi_k)$  is already sufficient for proving optimality, by Lemma 6.4. It is clear that having a larger basis without offering a stronger bound is inefficient, since solving  $P_\phi$  requires more time for larger matrices.

The example of  $\phi'_k$  and  $\phi_k$  shows that not all monomials are (equally) useful in determining bounds. It also shows that resolution can decrease the effectiveness of the SOS approach to the MAX-SAT problem, by providing ‘bad’ monomial bases, or it can occur that the  $SOS_p$  basis misses ‘good’ monomials. Our proposed basis  $SOS_p^Q$ , see (6.28), attempts to solve this issue.

## 6.6 Relating sum of squares and method of moments

In this section, we show how the SOS-SDP relaxation of van Maaren et al. [298, 299] and moment relaxations of Anjos [9, 10, 11, 12] are related. The relaxations of Anjos, as described in Section 6.3, were first introduced in 2004 [9] and can be considered as extensions of the GAP relaxation via the well-known Lasserre hierarchy [175]. In 2005, van Maaren et al. [298, 299] proposed the SOS approach to the (MAX-)SAT problem. Subsequently, van Maaren et al. [299] showed that the SOS relaxation, using monomial basis  $SOS_{pt}$  that is larger than  $SOS_p$ , see (6.27), outperforms the  $R_3$  relaxation of Anjos [10], in deciding on the satisfiability of 3-SAT instances. The  $R_3$  relaxation is known to dominate the  $R_2$  relaxation, see (6.11). In 2007, Anjos [12] strengthened his  $R_3$  relaxation further and left it as future work to determine which SDP relaxation was the strongest.

This chapter completes that work, by showing a simple relation between the two approaches. In particular, Anjos’ relaxations can be considered as method of moments

in the Lasserre hierarchy. It is well known that the method of moments is dual to SOS optimization, see [175], and we provide the details here. Let us first derive the dual of the SOS program  $P_\phi$ , see Theorem 6.6, and then relate it to the here proposed strengthened version of Anjos' relaxations.

To this end, we require the following intermediate result on  $v_\phi$ , see (6.6).

**Lemma 6.5.** *Let  $\phi = \bigwedge_{j=1}^m C_j$  be a logical proposition,  $v_\phi = \sum_{\alpha \subseteq [n]} v_\phi^\alpha x^\alpha$ , see (6.6), and  $F_\phi = \sum_{\alpha \subseteq [n]} p_\phi^\alpha x^\alpha$ , see (6.24). Then,  $v_\phi = m - F_\phi$ , and  $v_\phi^\alpha = -p_\phi^\alpha$  for all nonempty  $\alpha \subseteq [n]$ .*

*Proof.* Let clause  $C_j$  have length  $\ell_j$ . We have  $v(C_j) = 1 - 2^{-\ell_j} \prod_{i \in C_j} (1 - a_{j,i} x_i)$ , see (6.4). Then

$$v_\phi := \sum_{j \in [m]} v(C_j) = m - \sum_{j \in [m]} \frac{1}{2^{\ell_j}} \left( \sum_{\gamma \subseteq C_j} (-1)^{|\gamma|} a_j^\gamma x^\gamma \right) = m - F_\phi.$$

□

Let  $\mathbf{x}$  be a given monomial basis,  $S \subseteq \mathcal{S}^{|\mathbf{x}|}$ . Matrix  $S$  is indexed by all  $\alpha \subseteq [n]$  for which  $x^\alpha \in \mathbf{x}$ . To simplify the comparison between the SOS approach and the relaxations of Anjos [9], we define the set

$$\mathcal{X}_\phi := \left\{ S \in \mathcal{S}^{|\mathbf{x}|} : \begin{array}{l} \text{diag}(S) = \mathbf{1}, S_{\alpha,\beta} = S_{\alpha',\beta'} \quad \forall (\alpha,\beta,\alpha',\beta') \subseteq [n] \\ \text{such that } \alpha \Delta \beta = \alpha' \Delta \beta' \end{array} \right\}, \quad (6.44)$$

for a proposition  $\phi$  on  $n$  variables and  $m$  clauses. Note that  $\mathcal{X}_\phi \subseteq \bigcap_{j \in [m]} \Delta_j$ , see (6.10), since  $\Delta_j$  only restricts entries  $S_{\alpha,\beta}$  whenever  $\alpha$  and  $\beta$  are jointly contained in a single clause. We use  $\mathcal{X}_\phi$  in the following theorem.

**Theorem 6.6.** *Let  $\phi$  be a logical proposition and  $\mathbf{x}$  a monomial basis. The SOS program  $P_\phi$  defined by  $\phi$  and  $\mathbf{x}$ , is equivalent to*

$$\max_S \langle C, S \rangle \quad \text{s.t.} \quad S \in \mathcal{X}_\phi \cap \mathcal{S}_+, \quad (6.45)$$

where  $\mathcal{X}_\phi$  is given by (6.44) and  $C \in \mathcal{S}^{|\mathbf{x}|}$ , indexed by the subsets  $\alpha \subseteq [n]$  for which  $x^\alpha \in \mathbf{x}$ , is any matrix that satisfies

$$\sum_{(\alpha,\beta) \in \mathbf{x}^\gamma} C_{\alpha,\beta} = v_\phi^\gamma, \quad \forall \gamma \neq \emptyset, \mathbf{x}^\gamma \neq \emptyset$$

for  $v_\phi^\gamma$  as in (6.6). Moreover, (6.45) is strictly feasible.

*Proof.* We rewrite program  $P_\phi$  by splitting the matrix variable  $M$  as follows

$$v := \min_{M,Z} \langle \mathbf{I}, M \rangle \quad \text{s.t.} \quad Z \in \mathcal{S}_+, M \in \mathcal{M}_\phi, Z = M, \quad (6.46)$$

where  $\mathcal{M}_\phi$  is given in (6.32). We dualize the constraint  $M = Z$ , and set

$$g(S) := \min_{M \in \mathcal{M}_\phi, Z \succeq 0} \langle \mathbf{I}, M \rangle + \langle S, M - Z \rangle,$$

for some  $S \in \mathcal{S}$ . Clearly,  $g(S) \leq v$  for all  $S$ , and we thus look to maximize  $g(S)$ , i.e.,

$$\begin{aligned} \max_S g(S) &= \max_S \left[ \min_{M \in \mathcal{M}_\phi} \langle \mathbf{I} + S, M \rangle + \min_{Z \geq 0} \langle S, -Z \rangle \right] \\ &= \max_{S \geq 0} \min_{M \in \mathcal{M}_\phi} \langle \mathbf{I} + S, M \rangle = \max_{S \geq 0} \min_{M \in \mathcal{M}_\phi} \langle \mathbf{I} - S, M \rangle. \end{aligned} \quad (6.47)$$

We now determine the set  $\mathcal{X}_\phi$  such that, whenever  $S \in \mathcal{X}_\phi$ , the minimization over  $M \in \mathcal{M}_\phi$  in (6.47) is bounded. Observe that  $\mathcal{M}_\phi$  places no restrictions on the diagonal. To guarantee a bounded minimum, set  $\mathcal{X}_\phi$  should restrict  $\text{diag}(\mathbf{I} - S) = \mathbf{0}$ . Each off-diagonal element of a matrix in  $\mathcal{M}_\phi$  is restricted by a single constraint of the form (6.31). Therefore, solving (6.47) for  $M$  can be done by considering separately the elements of  $M$  restricted by a single constraint. That is,

$$\min_{M \in \mathcal{S}} \sum_{\gamma \in \mathbf{X}} \sum_{(\alpha, \beta) \in \mathbf{x}^\gamma} -S_{\alpha, \beta} M_{\alpha, \beta} \quad \text{s.t.} \quad \sum_{(\alpha, \beta) \in \mathbf{x}^\gamma} M_{\alpha, \beta} = p_\phi^\gamma,$$

where  $\mathbf{x}^\gamma$  and  $\mathbf{X}$  are defined in (6.30) and (6.26), respectively. This minimization problem is bounded if and only if

$$S_{\alpha, \beta} = S_{\alpha', \beta'}, \quad \forall (\alpha, \beta), (\alpha', \beta') \in \mathbf{x}^\gamma \quad \forall \mathbf{x}^\gamma \in \mathbf{X}, \quad (6.48)$$

or equivalently,  $S_{\alpha, \beta} = S_{\alpha', \beta'}$  for all possible index pairs  $(\alpha, \beta)$  and  $(\alpha', \beta')$  that satisfy  $\alpha \Delta \beta = \alpha' \Delta \beta'$ . It follows that  $\mathcal{X}_\phi$  is given by (6.44). Now, for fixed  $S \in \mathcal{X}_\phi \cap \mathcal{S}_+$ , any matrix  $M \in \mathcal{M}_\phi$  obtains the same value in (6.47). Note also that w.l.o.g., we may fix  $M = \mathcal{P}_{\mathcal{M}_\phi}(\mathbf{0})$ , i.e., the projection of the zero matrix onto  $\mathcal{M}_\phi$ , (see Lemma 6.7) which has zero diagonal. This yields the equivalent program of the form (6.45), for  $C = -M = -\mathcal{P}_{\mathcal{M}_\phi}(\mathbf{0})$ . Written explicitly,

$$C_{\alpha, \beta} = -\frac{p_\phi^\gamma}{|\mathbf{x}^\gamma|}, \quad \forall \alpha, \beta \subseteq [n] \text{ such that } \alpha \Delta \beta = \gamma \text{ (i.e., } (\alpha, \beta) \in \mathbf{x}^\gamma).$$

This combined with Lemma 6.5, proves the claim on matrix  $C$ . Lastly, observe that the identity matrix of appropriate size is strictly feasible for (6.45).  $\square$

We define, for  $S \in \mathcal{X}_\phi$  and each clause  $C_j$ , the function  $v^{\text{SDP}}(S, C_j)$ , which is obtained by taking (6.4), and replacing each  $x^\gamma$  by  $S_{\alpha, \beta}$ , for some  $(\alpha, \beta) \in \mathbf{x}^\gamma$ . By (6.48), we are allowed to pick any such  $(\alpha, \beta)$ . By Lemma 6.5, for any nonempty  $\gamma \subseteq [n]$ ,  $S \in \mathcal{X}_\phi$  and  $C$  as in Theorem 6.6, we have

$$\sum_{(\alpha, \beta) \in \mathbf{x}^\gamma} C_{\alpha, \beta} S_{\alpha, \beta} = \sum_{(\alpha, \beta) \in \mathbf{x}^\gamma} \frac{-p_\phi^\gamma}{|\mathbf{x}^\gamma|} S_{\alpha, \beta} = -p_\phi^\gamma S_{\alpha, \beta} = v_\phi^\gamma S_{\alpha, \beta}.$$

Hence, maximizing  $\langle C, S \rangle$  is equivalent to maximizing the semidefinite relaxation of  $v_\phi$ , see (6.6), which equals  $\sum_{j \in [m]} v^{\text{SDP}}(S, C_j)$ .

Moreover, in the relaxations of Anjos [9, 10, 11, 12], outlined in Section 6.3, the matrix variable is restricted to satisfy  $v^{\text{SDP}}(S, C_j) = 1$ . Now we can easily observe the difference between the SOS-SDP relaxations and those proposed by Anjos. We

present the equivalent dual formulation of the SOS approach below on the left-hand side and the latter (in slightly adapted form) on the right.

$$\begin{aligned}
 v^* = \max \quad & \sum_{j \in [m]} v^{\text{SDP}}(S, C_j) & \max \quad & 0 \\
 \text{s.t.} \quad & S \in \mathcal{X}_\phi \cap \mathcal{S}_+ & \text{s.t.} \quad & S \in \mathcal{X}_\phi \cap \mathcal{S}_+ \\
 & & & v^{\text{SDP}}(S, C_j) = 1, \forall C_j.
 \end{aligned} \tag{6.49} \tag{6.50}$$

Note again the difference between (6.50) and the relaxations described in Section 6.3, resulting from using set  $\mathcal{X}_\phi$  instead of the intersection of the  $\Delta_j$ , see (6.10). Thus, we compare the SOS approach with a strengthened variant of the relaxation proposed by Anjos. In Section 6.8.3, we determine the dual of (6.50).

Program (6.49) proves unsatisfiability of  $\phi$  if  $v^* < m$  (with some margin of error, due to numerical precision), while (6.50) does so whenever the program is infeasible. The programs are not equivalent in this sense: we have empirically found instances  $\phi$  for which  $v^* \geq m$ , while (6.50) is infeasible. Neither program can directly prove satisfiability. However, solutions to both programs can be used to guide the search towards satisfying assignments (should they exist), see Section 6.7.3.

If (6.50) admits a feasible matrix  $S^*$ , then matrix  $S^*$  is clearly also feasible for (6.49) and attains an objective value of  $m$ . Consequently, in this case, we have  $v^* \geq m$ . Thus, if (6.50) does not prove unsatisfiability of  $\phi$ , then neither does (6.49). In Section 6.10 we show that (6.49) can be computed efficiently by applying the PRSM to its dual<sup>1</sup>. It is currently unclear whether a good algorithm for solving (6.50) exists, and if so, how efficient it would be. Previous numerical experiments on (6.50) have used general purpose SDP solvers. An immediate improvement might be to use an SDP feasibility problem solver, see [77, 138].

Lastly, the objective value of (6.49) is more useful for the MAX-SAT problems: if the underlying instance is infeasible,  $v^*$  provides an upper bound to the number of satisfiable clauses, which is useful in a B&B scheme. Program (6.50) might also show unsatisfiability of the same instance, but its infeasibility offers no additional value, as to *how* unsatisfiable the instance is.

## 6.7 The Peaceman-Rachford splitting method for the MAX-SAT problem

In this section, we introduce the *Peaceman-Rachford splitting method* [251] for solving SDPs and apply it to the MAX-SAT SOS program  $P_\phi$ . Conventionally, interior-point methods are used to solve SDPs. However, for medium and large size instances, interior-point methods suffer from a large computation time and memory demand, which has recently motivated researchers to consider first-order methods, such as the PRSM. For recent applications of PRSM to SDP, see e.g., [69, 121].

Sections 6.7.2 and 6.7.3 provide details on obtaining valid upper and lower bounds, from the output of the PRSM algorithm.

---

<sup>1</sup>Program (6.49) can also be directly solved with the PRSM, as projecting onto  $\mathcal{X}_\phi$  is computationally cheap.

### 6.7.1 The PRSM for SOS relaxations of the MAX-SAT problem

We start from the reformulation of  $P_\phi$  given in (6.46). The augmented Lagrangian function of (6.46) w.r.t. the constraint  $M = Z$  for a penalty parameter  $\rho > 0$  is:

$$\mathcal{L}_\rho(Z, M, S) = \langle \mathbf{I}, M \rangle + \langle S, M - Z \rangle + \frac{\rho}{2} \|M - Z\|^2.$$

Here,  $S \in \mathcal{S}$  is the Lagrange multiplier and  $\|\cdot\|$  denotes the Frobenius matrix norm, i.e.,  $\|X\| := \sqrt{\langle X, X \rangle}$  for  $X \in \mathcal{S}$ .

The PRSM now entails iteratively optimizing over the variables  $Z$  and  $M$  separately, and updating  $S$  twice per cycle. We write superscript  $k$  to denote the value of the variable at iteration  $k$ .

$$\begin{cases} Z^{k+1} = \arg \min_{Z \succeq 0} \mathcal{L}_\rho(Z, M^k, S^k) & = \mathcal{P}_{\mathcal{S}_+} \left( M^k + \frac{1}{\rho} S^k \right) \\ S^{k+\frac{1}{2}} = S^k + \nu_1 \rho (M^k - Z^{k+1}) \\ M^{k+1} = \arg \min_{M \in \mathcal{M}_\phi} \mathcal{L}_\rho(Z^{k+1}, M, S^{k+\frac{1}{2}}) & = \mathcal{P}_{\mathcal{M}_\phi} \left( Z^{k+1} - \frac{1}{\rho} [\mathbf{I} + S^{k+\frac{1}{2}}] \right) \\ S^{k+1} = S^{k+\frac{1}{2}} + \nu_2 \rho (M^{k+1} - Z^{k+1}). \end{cases} \quad (6.51)$$

Here,  $\mathcal{M}_\phi$  is as in (6.32), and  $\mathcal{P}$  is the orthogonal projection operator. That is, for a closed and convex set  $\mathcal{F} \subseteq \mathcal{S}$ , and some  $X \in \mathcal{S}$ ,

$$\mathcal{P}_{\mathcal{F}}(X) := \arg \min_{Y \in \mathcal{F}} \|X - Y\|.$$

For the second equality in (6.51), we have used that

$$\begin{aligned} \arg \min_{Z \succeq 0} \mathcal{L}_\rho(Z, M, S) &= \arg \min_{Z \succeq 0} \langle \mathbf{I}, M \rangle - \frac{1}{2\rho} \|S\|^2 + \frac{\rho}{2} \left\| Z - \left( M + \frac{1}{\rho} S \right) \right\|^2 \\ &= \arg \min_{Z \succeq 0} \frac{\rho}{2} \left\| Z - \left( M + \frac{1}{\rho} S \right) \right\|^2, \end{aligned}$$

see e.g., [243]. In an implementation of (6.51), one should not store matrix  $S^k$  directly, but rather, the matrix  $\frac{1}{\rho} S^k$ , see Appendix A.1.

When a symmetric matrix  $X \in \mathcal{S}$  has eigenvalues  $\lambda_i$ , and corresponding orthonormal eigenvectors  $v_i$ , it is well known that the projection onto the positive semidefinite cone is given by

$$\mathcal{P}_{\mathcal{S}_+}(X) = \sum_{i:\lambda_i > 0} \lambda_i v_i v_i^\top = X - \sum_{i:\lambda_i < 0} \lambda_i v_i v_i^\top. \quad (6.52)$$

Depending on the number of positive eigenvalues of  $X$ , one of the expressions in (6.52) will be cheaper to compute. The next lemma shows how to compute a projection onto  $\mathcal{M}_\phi$ . A similar result is also provided by [254, Prop. 7].

**Lemma 6.7.** *Let  $M \in \mathcal{S}$ , indexed by a collection of subsets of  $[n]$ . Consider  $\widehat{M} := \mathcal{P}_{\mathcal{M}_\phi}(M)$ . We have that  $\text{diag}(\widehat{M}) = \text{diag}(M)$  and*

$$\widehat{M}_{\delta,\mu} = M_{\delta,\mu} - \frac{1}{|\mathbf{x}^\gamma|} \left( \sum_{(\alpha,\beta) \in \mathbf{x}^\gamma} M_{\alpha,\beta} - p_\phi^\gamma \right), \quad (6.53)$$

for  $(\delta, \mu) \in \mathbf{x}^\gamma$ , see (6.30), with  $\gamma \neq \emptyset$ . In particular, when  $|\mathbf{x}^\gamma| = 2$ , (6.53) reduces to

$$\widehat{M}_{\delta,\mu} = \widehat{M}_{\mu,\delta} = p_\phi^\gamma/2. \quad (6.54)$$

*Proof.* Let  $M \in \mathcal{S}$ . To find  $\widehat{M} = \mathcal{P}_{\mathcal{M}_\phi}(M)$ , note that in  $\mathcal{M}_\phi$ , each off-diagonal entry is restricted by exactly one constraint. This follows from (6.32). Since  $\mathcal{M}_\phi$  does not restrict the diagonal, it is easily seen that  $\text{diag}(\widehat{M}) = \text{diag}(M)$ . Now for the off-diagonal entries, we fix a nonempty  $\gamma \subseteq [n]$  and define  $\mathbf{m}$  as the vector that contains upper triangular entries of  $M$ ,  $M_{\alpha,\beta}$ , such that  $(\alpha, \beta) \in \mathbf{x}^\gamma$ . Similarly, we define  $\widehat{\mathbf{m}}$  as the vector containing the same entries of matrix  $\widehat{M}$ , rather than  $M$ . Note that  $\mathbf{1}^\top \widehat{\mathbf{m}} = p_\phi^\gamma/2$ . Minimizing the Frobenius norm of  $\widehat{M} - M$  is now equivalent to minimizing the norm of  $\widehat{\mathbf{m}} - \mathbf{m}$ . Thus, we solve

$$\widehat{\mathbf{m}} = \arg \min_{\mathbf{1}^\top v = p_\phi^\gamma/2} \|v - \mathbf{m}\|^2,$$

which can be done analytically and leads to (6.53). The simplification of (6.53) to (6.54) follows from the equality  $\sum_{(\alpha,\beta) \in \mathbf{x}^\gamma} M_{\alpha,\beta} = 2M_{\delta,\mu}$ , whenever  $|\mathbf{x}^\gamma| = 2$  and  $(\delta, \mu) \in \mathbf{x}^\gamma$ .  $\square$

Due to the presence of many unit constraints, see (6.33), these projections are computationally cheap to compute, and hence, the PRSM is well suited to exploit this. Lastly, it is proven [136] that (6.51) converges for  $(\nu_1, \nu_2) \in \mathcal{D}$ , where

$$\mathcal{D} := \left\{ (\nu_1, \nu_2) \in \mathbb{R}^2 : \begin{array}{l} |\nu_1| < \min\{1, 1 + \nu_2 - \nu_2^2\}, \\ 0 < \nu_2 < \frac{1+\sqrt{5}}{2}, \nu_1 + \nu_2 > 0 \end{array} \right\},$$

The values that we choose for  $(\nu_1, \nu_2)$ , and other parameters, are given in Section 6.10.

## 6.7.2 Upper bounds, lower bounds and early stopping

After each PRSM iteration  $k$  we obtain a triple  $(Z^k, M^k, S^k)$  and the value  $\langle \mathbf{I}, M^k \rangle$ . Although this value converges to the optimal objective value of the SDP  $\mathbf{P}_\phi$ , the convergence is typically not monotonic and therefore this value does not necessarily provide a valid upper bound for the problem. In this section we describe how to obtain a valid upper bound from the output of the PRSM.

Observe that the feasible set of  $\mathbf{P}_\phi$  depends on the chosen monomial  $\mathbf{x}$  through  $\mathcal{V}_\mathbf{x}$ , see (6.22). Hence, by (6.23), we have

$$p_\phi^\emptyset - \min_{M \in \mathcal{M}_\phi \cap \mathcal{S}_+} \langle \mathbf{I}, M \rangle = \sup_{\mu \in \mathbb{R}} \{ \mu : F_\phi - \mu \in \mathcal{V}_\mathbf{x} \} \leq \min_{x \in \{\pm 1\}^n} F_\phi, \quad (6.55)$$

for  $p_\phi^\emptyset$  as in (6.25). From (6.55) it follows that the maximum number of satisfiable clauses of  $\phi$  is bounded from above by

$$m - p_\phi^\emptyset + \min_{M \in \mathcal{M}_\phi \cap \mathcal{S}_+} \langle \mathbf{I}, M \rangle, \quad (6.56)$$

for  $m$  equal to the number of clauses in  $\phi$ . Since the number of satisfied clauses is an integer, the bound (6.56) can be improved by rounding down the result.

Ideally, the PRSM algorithm (6.51) computes the upper bound (6.56) by finding an optimal  $M$  in the set  $\mathcal{M}_\phi \cap \mathcal{S}_+$ . However, in practice one terminates the PRSM algorithm before such an optimal  $M$  has been found. Let matrix  $M^k$  then be defined as in (6.51) and let  $\lambda_{\min}(M^k)$  be its smallest eigenvalue. Note that

$$\widetilde{M}^k = M^k - \lambda_{\min}(M^k) \mathbf{I} \in \mathcal{M}_\phi \cap \mathcal{S}_+, \quad (6.57)$$

and so,  $\widetilde{M}^k$  is feasible for  $\mathbf{P}_\phi$ . Thus, a valid upper bound at iteration  $k$  is obtained as follows:

$$\left\lfloor m - p_\phi^\emptyset + \langle \mathbf{I}, \widetilde{M}^k \rangle \right\rfloor. \quad (6.58)$$

### 6.7.3 Lower bounds and rounding

In order to obtain a truth assignment of the variables from the output of the PRSM one needs a rounding procedure. We describe here the rounding procedure proposed by van Maaren et al. [299] and a modification of the procedure that is implemented in our solver.

Let matrix  $M^*$  be the optimal solution to the SOS program  $\mathbf{P}_\phi$ , induced by a logical proposition  $\phi$  on  $n$  variables. Let  $\mathbf{x}$  be its monomial basis of size  $s$ , and  $\mu^*$  such that  $F_\phi(x) - \mu^* \equiv \mathbf{x}^\top M^* \mathbf{x} \pmod{\mathcal{I}}$ . It is clear that, by optimality of  $M^*$ ,  $\lambda_{\min}(M^*) = 0$ . Let  $N$  be the multiplicity of the zero eigenvalue, and  $\mathbf{v}_i$ ,  $i \in [N]$  the corresponding eigenvectors. If  $y \in \{\pm 1\}^n$  is an optimal MAX-SAT truth assignment of  $\phi$ , then  $y$  minimizes  $F_\phi$ . Let  $y'$  be the monomial basis vector  $\mathbf{x}$ , evaluated with the entries of  $y$ . Then  $(y')^\top M^* y' = F_\phi(y) - \mu^*$ . If the SOS relaxation  $\mathbf{P}_\phi$  computes the optimal bound, we have  $\mu^* = F_\phi(y)$ , which implies that  $(y')^\top M^* y' = 0$ . As the eigenvectors  $\mathbf{v}_i$  satisfy the same relation, i.e.  $(\mathbf{v}_i)^\top M^* \mathbf{v}_i = 0$ , they can be considered as approximations of maximally satisfying assignments.

Let  $V \in \mathbb{R}^{s \times N}$  be the matrix having the vectors  $\mathbf{v}_i$ ,  $i \in [N]$  as columns. Each row of  $V$  corresponds to a monomial of  $\mathbf{x}$ . For  $p$  the number of monomials in  $\mathbf{x}$  of degree two or more, matrix  $B \in \mathbb{R}^{p \times N}$  is the submatrix of  $V$  obtained by taking the rows of  $V$  corresponding to these  $p$  monomials. We define  $U \in \mathbb{R}^{N \times N}$  as the matrix with columns the eigenvectors of  $B^\top B$ .

The rounding procedure proposed by van Maaren et al. is to compute  $x_\lambda \in \{\pm 1\}^n$  as

$$x_\lambda = \text{sgn}(P_1) \begin{bmatrix} \mathbf{0}_{n \times 1} & \mathbf{I}_n & \mathbf{0}_{n \times (s-n-1)} \end{bmatrix} \text{sgn}(P), \quad (6.59)$$

for  $P = VU\tilde{\lambda}$  and  $\tilde{\lambda}_i = \xi_i \lambda_i \forall i \in [N]$ .

Here,  $\lambda \in \mathbb{R}^N$  is a vector generated uniformly at random on the unit sphere (which allows us to perform multiple roundings by generating multiple  $\lambda$ ). Observe that  $P_1$ ,

the first entry of vector  $P$ , corresponds to the monomial  $x^\emptyset$ . The vector  $\xi \in \mathbb{R}^N$  is a parameter to be chosen. We refer to [299] for the details.

The optimal matrix  $M^*$  is a low rank matrix, which ensures that  $N$ , the multiplicity of eigenvalue 0, satisfies  $N > 1$ . In practice however, we do not find  $M^*$ , but its approximation  $\widetilde{M}^k$  at some iteration, see (6.57). Due to early stopping, matrix  $\widetilde{M}^k$  often has eigenvalue 0 with multiplicity 1. Then  $\lambda$  is a scalar, which does not affect (6.59). Thus, when  $N = 1$ , we can only perform one rounding. To solve this issue, we propose constructing  $V$  with the columns of  $q$  eigenvectors corresponding to the  $q$  smallest eigenvalues of  $\widetilde{M}^k$  (only in case  $N < q$ ). Thus, whenever  $N < q$ , we add  $q - N$  eigenvectors corresponding to nonzero eigenvalues to the matrix  $V$ , in order to perform multiple roundings. In [299], it is observed that the rounding procedure works better when  $N$  is small. We use this information by setting  $q = 4$ , so that we take at least four vectors for the rounding procedure.

## 6.8 The weighted partial MAX-SAT problem

In this section, we extend the SOS approach to the MAX-SAT problem, to the weighted partial MAX-SAT problem. We also show that the dual formulation of the SOS program for certain partial MAX-SAT instances, equals the relaxations by Anjos [10].

In the *weighted* MAX-SAT problem, each clause is given a weight, and the objective is to maximize the sum of the weights of the satisfied clauses. In the *partial* MAX-SAT problem, clauses are divided in soft and hard clauses. The aim is to maximize the number of satisfied soft clauses, while satisfying all the hard clauses. The combination of the weighted and partial MAX-SAT problems is clear, and referred to as the *weighted partial* MAX-SAT problem [195].

Consider again a logical proposition  $\phi$ . Let  $w_j \in \mathbb{R}$  be the weight associated to clause  $C_j$ . The generalization of (6.19) for the (unweighted) MAX-SAT problem, to the weighted MAX-SAT problem follows by setting

$$F_\phi^{\mathcal{W}}(x) = \sum_{j=1}^m \frac{w_j}{2^{\ell_j}} \prod_{i \in C_j} (1 - a_{j,i} x_i), \quad (6.60)$$

and then minimizing  $F_\phi^{\mathcal{W}}$  for  $x \in \{\pm 1\}^n$ . This minimization can be approximated by SOS optimization, using directly the SDP  $\mathbf{P}_\phi$ .

For the weighted partial MAX-SAT problem, consider a logical proposition  $\phi$ , on  $n$  variables,  $m$  soft clauses  $C_j$  and  $q$  hard clauses  $C_p^{\mathbf{H}}$ . To each hard clause  $C_p^{\mathbf{H}}$ ,  $p \in [q]$ , we associate the polynomial  $f_p = \prod_{i \in [n]} (1 - a_{p,i} x_i)$ , similar to (6.60). Note that  $f_p$  vanishes for all truth assignments that satisfy clause  $C_p^{\mathbf{H}}$ . Similar to (6.20)

and (6.22), we define the sets

$$\begin{aligned} \mathcal{H}_{\mathbf{x}} &:= \left\{ \sum_{p \in [q]} c_p f_p \pmod{\mathcal{I}} : c_p \in \mathbb{R} \ \forall p \in [q] \right\} \\ &\subseteq \mathcal{H} := \left\{ \sum_{p \in [q]} g_p f_p \pmod{\mathcal{I}} : g_p \in \mathbb{R}[x] \ \forall p \in [q] \right\}. \end{aligned} \quad (6.61)$$

Let  $\mathbf{SAT} \subseteq \{\pm 1\}^n$  be the set of all truth assignments that satisfy the hard clauses (which we assume to be nonempty). From [261] it follows that

$$\begin{aligned} \min_{x \in \mathbf{SAT}} F_{\phi}^{\mathcal{W}}(x) &= \sup \{ \mu : F_{\phi}^{\mathcal{W}} - \mu \in \mathcal{V} + \mathcal{H} \} \\ &\geq \sup \{ \mu : F_{\phi}^{\mathcal{W}} - \mu \in \mathcal{V}_{\mathbf{x}} + \mathcal{H}_{\mathbf{x}} \}, \end{aligned} \quad (6.62)$$

where  $\mathcal{V}$  is as in (6.20), and ‘+’ denotes the Minkowski sum of sets. We proceed by writing the lower bound in (6.62) as an explicit SDP, for which we introduce the following sets

$$\mathbf{H}^{\gamma} := \{ p \in [q] : \gamma \subseteq C_p^{\mathbf{H}} \} \quad \gamma \subseteq [n].$$

Set  $\mathbf{H}^{\gamma}$  contains all  $p$  for which  $f_p$ , when expanded, contains the term  $\pm x^{\gamma}$ . The sign here is determined by the parity of  $|\gamma \cap I_p^+|$ , see (6.2). Additionally, we define as analogue to  $\mathcal{M}_{\phi}$ , see (6.32), the set  $\mathcal{M}_{\phi}^{\mathbf{H}}$ . This set contains all matrices  $M$  and vectors  $c$  such that  $F_{\phi}^{\mathcal{W}} - \mu \equiv \mathbf{x}^{\top} M \mathbf{x} + \sum_{p \in [q]} c_p f_p \pmod{\mathcal{I}}$ . It is therefore defined as

$$\mathcal{M}_{\phi}^{\mathbf{H}} := \left\{ (M, c) \in \mathcal{S} \times \mathbb{R}^q : \begin{array}{l} \forall \gamma \neq \emptyset \text{ s.t. } x^{\gamma} \in \mathbf{X}, \sum_{(\alpha, \beta) \in \mathbf{x}^{\gamma}} M_{\alpha, \beta} \\ + \sum_{p \in \mathbf{H}^{\gamma}} (-1)^{|\gamma \cap I_p^+|} c_p = p_{\phi}^{\gamma} \end{array} \right\}. \quad (6.63)$$

This allows us to adapt  $P_{\phi}$  to the weighted partial MAX-SAT problem as follows:

$$\min_{M, c} \langle \mathbf{I}, M \rangle + \sum_{p \in [q]} c_p \quad \text{s.t.} \quad (M, c) \in \mathcal{M}_{\phi}^{\mathbf{H}}, \quad M \in \mathcal{S}_+. \quad (6.64)$$

We approximately solve (6.64) by the PRSM, see Section 6.8.1.

Let us elaborate on how to adapt the monomial bases to the (weighted) partial MAX-SAT problem. We make no distinction between soft and hard clauses for the  $SOS_p$  basis, see (6.27). For basis  $SOS_s^{\theta}$ , see (6.29), we determine the variable weights as  $w(i) := \sum_{\{j: i \in C_j\}} w_j + \sum_{\{p: i \in C_p^{\mathbf{H}}\}} \bar{w}$ , for  $\bar{w}$  the mean of all soft clause weights  $w_j$ . For basis  $SOS_p^{\mathbf{Q}}$ , we add all  $\binom{\mathbf{Q}}{2}$  quadratic terms of the  $\mathbf{Q}$  variables that attain the largest value of  $w(i)$ . For unweighted partial MAX-SAT instances, we consider all  $w_j$  to equal 1.

### 6.8.1 The PRSM for SOS of the weighted partial MAX-SAT problem

We show here how to solve (6.64) by the PRSM. We first rewrite (6.64) by introducing the matrix variable  $Z$ , see also (6.46),

$$\begin{aligned} \min \quad & \langle \mathbf{I}, M \rangle + \sum_{p \in [q]} c_p \\ \text{s.t.} \quad & Z \in \mathcal{S}_+, (M, c) \in \mathcal{M}_\phi^{\mathbf{H}}, Z = M. \end{aligned} \quad (6.65)$$

Then, the augmented Lagrangian function of (6.65) w.r.t.  $Z = M$  and for a penalty parameter  $\rho > 0$  is:

$$\mathcal{L}_\rho(Z, M, S, c) = \langle \mathbf{I}, M \rangle + \mathbf{1}^\top c + \langle S, M - Z \rangle + \frac{\rho}{2} \|M - Z\|^2,$$

where  $S$  is the dual variable. The PRSM is iteratively and separately optimizing over  $(M, c) \in \mathcal{M}_\phi^{\mathbf{H}}$  and  $Z \in \mathcal{S}_+$ , and updating  $S$  twice per cycle, similarly to (6.51). However, in this case, the  $M$ -subproblem in the third line of (6.51) is replaced by the  $(M, c)$ -subproblem.

We now show that minimization over  $(M, c) \in \mathcal{M}_\phi^{\mathbf{H}}$  can be performed efficiently. By derivations similar to, e.g., [243, Eq. 3.4], we have:

$$\arg \min_{(M, c) \in \mathcal{M}_\phi^{\mathbf{H}}} \mathcal{L}_\rho(Z, M, S, c) = \arg \min_{(M, c) \in \mathcal{M}_\phi^{\mathbf{H}}} \left\| M - \tilde{Z} \right\|^2 + \frac{2}{\rho} \mathbf{1}^\top c, \quad (6.66)$$

where  $\tilde{Z} := Z - (S + \mathbf{I}) / \rho$ . This is a convex quadratic program (QP) that we solve in two steps. Firstly, consider the matrix-entries  $M_{\alpha, \beta}$ , with  $(\alpha, \beta) \in \mathbf{x}^\gamma$ , see (6.30), and  $\mathbf{H}^\gamma = \emptyset$ . Since  $\mathbf{H}^\gamma = \emptyset$ , these entries are unaffected by the  $c_p$  variables. This implies that these  $M_{\alpha, \beta}$  variables are not coupled with the other entries of  $M$ , and one can minimize separately over such  $M_{\alpha, \beta}$ . This separate minimization problem can be solved by applying Lemma 6.7.

Secondly, the remaining QP

$$\min \sum_{\gamma: \mathbf{H}^\gamma \neq \emptyset} \sum_{(\alpha, \beta) \in \mathbf{x}^\gamma} \left( M_{\alpha, \beta} - \tilde{Z}_{\alpha, \beta} \right)^2 + \frac{2}{\rho} \mathbf{1}^\top c, \quad (6.67)$$

can be simplified by the following observation. If  $M^*$  is an optimal solution to (6.66), then

$$(\alpha, \beta), (\alpha', \beta') \in \mathbf{x}^\gamma \implies M_{\alpha, \beta}^* - \tilde{Z}_{\alpha, \beta} = M_{\alpha', \beta'}^* - \tilde{Z}_{\alpha', \beta'}.$$

Hence, (6.67) can be simplified by substituting each term  $\sum_{(\alpha, \beta) \in \mathbf{x}^\gamma} (M_{\alpha, \beta} - \tilde{Z}_{\alpha, \beta})^2$  with a single squared variable. We solve the resulting QP either by solving the linear KKT conditions using the LU decomposition, or via MOSEK [228]. The solving method depends on the underlying QP. Note that the KKT conditions define a linear system in which the associated matrix is indefinite, so that the Cholesky decomposition cannot be used.

### 6.8.2 Strengthening the bounds

We demonstrate a simple technique for improving the upper bounds given by program (6.64). This technique is based on the SAT resolution rule, which is given as follows. For two hard clauses of some proposition  $\phi$ , on literals  $x, z_i, i \in [s]$  and  $y_i, i \in [t]$ , construct the clause below the horizontal line:

$$\frac{[x \vee z_1 \vee \dots \vee z_s], [\neg x \vee y_1 \vee \dots \vee y_t]}{z_1 \vee \dots \vee z_s \vee y_1 \vee \dots \vee y_t}. \quad (6.68)$$

In contrast to the MAX-SAT resolution rule (6.41), the SAT resolution rule states that one may add the clause below the horizontal line to  $\phi$ , without changing its (un)satisfiability (we say that the new clause is *implied* by the original two clauses). We may apply this SAT resolution rule to the hard clauses of a partial MAX-SAT instance to generate more hard clauses. As each new clause induces a new variable  $c_p$ , the bound of program (6.64) can only improve. One may also regard SAT resolution as extending the set  $\mathcal{H}_x$ , see (6.61), by including terms of the form  $c_p x^\alpha f_p$ , for some  $\alpha \subseteq [n]$  where  $c_p \in \mathbb{R}$ .

Additionally, SAT resolution can generate hard unit clauses. This is advantageous, since hard unit clauses reduce the number of variables in the MAX-SAT problem as explained in Section 6.1.1.

### 6.8.3 Duality in the partial MAX-SAT problem

Now we consider partial MAX-SAT problems with only hard clauses. Solving such instances is thus equivalent to determining the satisfiability of the given hard clauses. We show that by taking the dual of the resulting SOS program, one obtains (a stronger version of) the relaxations of Anjos [9], given by (6.50).

We define, for  $A \in \mathcal{S}^n$ ,  $\text{vec}(A) \in \mathbb{R}^{n^2}$  the vector whose entries are the columns of  $A$  stacked together. We start from program (6.64) and perform variable splitting on  $M$ , similar to (6.46). We take the dual  $g(S)$  of this formulation, similar to (6.47), and consider the problem

$$\max_S g(S) = \max_{S \in \mathcal{X}_\phi \cap \mathcal{S}_+} \min_{(M,c) \in \mathcal{M}_\phi^{\mathbf{H}}} \langle S, -M \rangle + \mathbf{1}^\top c, \quad (6.69)$$

for  $\mathcal{X}_\phi$  as in (6.44). The steps that show that  $S \in \mathcal{X}_\phi \cap \mathcal{S}_+$  is necessary for (6.69) to be finite are provided in the proof of Theorem 6.6. We rewrite the inner minimization problem in (6.69) as

$$\min_{(M,c) \in \mathcal{M}_\phi^{\mathbf{H}}} \begin{bmatrix} \text{vec}(S) \\ \mathbf{1} \end{bmatrix}^\top \begin{bmatrix} \text{vec}(-M) \\ c \end{bmatrix}, \quad (6.70)$$

and proceed to show under which conditions this value is bounded. Observe that the coefficients  $p_\phi^\gamma = 0$ , see (6.63), since there are no soft clauses. Moreover, the set  $\mathcal{M}_\phi^{\mathbf{H}}$  places only linear constraints on the entries of  $M$  and  $c$ . Therefore, there exists a matrix  $D$  that satisfies

$$(M, c) \in \mathcal{M}_\phi^{\mathbf{H}} \iff D \begin{bmatrix} \text{vec}(-M) \\ c \end{bmatrix} = \mathbf{0}.$$

Hence, (6.70) is bounded if and only if  $[\text{vec}(S)^\top \quad \mathbf{1}^\top]$  is contained in the row space of  $D$ . This is precisely the requirement that  $v^{\text{SDP}}(S, C_p^{\mathbf{H}}) = 1, \forall p \in [q]$ , as in (6.50). We provide one example of this claim.

**Example 6.8.** Consider the monomial basis  $\mathbf{x} = (x^\emptyset, x_1, x_2)$ . Let  $C_1^{\mathbf{H}} = x_1 \vee x_2$ , so that  $f_1 = 1 - x_1 - x_2 + x_1x_2$ . Now

$$(M, c) \in \mathcal{M}_\phi^{\mathbf{H}} \implies Du = \mathbf{0}, \text{ for } D = \begin{bmatrix} -1 & -1 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & -1 & -1 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & -1 & -1 & 1 \end{bmatrix},$$

$$\text{and } u = [-M_{1,\emptyset} \quad -M_{\emptyset,1} \quad -M_{2,\emptyset} \quad -M_{\emptyset,2} \quad -M_{1,2} \quad -M_{2,1} \quad c_1]^\top.$$

For  $S \in \mathcal{X}_\phi$ , and by definition of  $\mathcal{X}_\phi$  (6.44), we have  $S_{1,\emptyset} = S_{\emptyset,1}$ , and similar equalities hold for all other related entries of matrix  $S$ . Thus, we may remove duplicate columns in  $D$ . We have

$$\begin{aligned} [S_{1,\emptyset} \quad S_{2,\emptyset} \quad S_{1,2} \quad 1] \in \text{row} \begin{bmatrix} -1 & 0 & 0 & -1 \\ 0 & -1 & 0 & -1 \\ 0 & 0 & -1 & 1 \end{bmatrix} \\ \implies S_{1,\emptyset} + S_{2,\emptyset} - S_{1,2} = 1 \implies v^{\text{SDP}}(S, C_1^{\mathbf{H}}) = 1. \end{aligned}$$

△

Thus, (6.70) is bounded if  $[\text{vec}(S)^\top \quad \mathbf{1}^\top] \in \text{row}(D)$ , in which case, the objective value equals zero. Hence, program (6.69) is equivalent to (6.50).

## 6.9 SOS-MS: Algorithm description

In this section, we elaborate on the algorithm behind our complete SOS-SDP based MAX-SAT solver, named SOS-MS. In particular, we outline the main parts of SOS-MS and provide a pseudocode, see Algorithm 4.

Consider a given MAX- $k$ -SAT instance,  $k \leq 3$ , and corresponding logical proposition  $\phi$ . SOS-MS uses the PRSM, see (6.51), to obtain an approximate solution  $\widetilde{M}$ , see (6.57), to  $\mathbb{P}_\phi$ . Then, using (6.58), SOS-MS determines an upper bound  $\text{UB}$  and a lower bound on the optimal value, by applying the rounding procedure from Section 6.7.3. The solver calls once, at the beginning, the CCLS<sup>2</sup> algorithm [210] for the MAX-SAT problem, to compute another lower bound. CCLS is a local search algorithm whose performance was one of the best among tested heuristic algorithms in the MSE-2016. We set  $\text{LB}$  as the maximum value among these two lower bounds.

In case  $\text{LB} = \text{UB}$ , we have proven optimality and the algorithm terminates. In case  $\text{LB} < \text{UB}$ , we branch on some variable  $x_i, i \in [n]$ , by assigning it either **true** or **false**. This resembles to performing unit resolution, see Section 6.1.1. We write  $\phi' = \text{unitRes}(\phi, i)$  to indicate that  $\phi'$  is the logical proposition obtained from  $\phi$  by setting  $x_i = 1$  (equivalently,  $x_i = \text{true}$ ). We use the same notation to indicate the logical proposition obtained from  $\phi$  by setting  $x_i = -1$  (equivalently,  $x_i = \text{false}$ ), i.e.,

<sup>2</sup>The CCLS algorithm is publicly available at <http://lcs.ios.ac.cn/~caisw/MaxSAT.html>.

$\phi' = \mathbf{unitRes}(\phi, -i)$ . To emphasize the difference between  $\phi$  and  $\phi'$ , in this section, we write  $n_\phi$  and  $m_\phi$  for the number of variables and clauses of  $\phi$ .

If we branch on  $x_i$ , we remove from the monomial basis  $\mathbf{x}$  all monomials  $x^\alpha$  that satisfy  $i \in \alpha$ . We remove from matrices  $Z^k$ ,  $M^k$  and  $S^k$ , that were obtained in the last relevant call of the PRSM, all rows and columns corresponding to such subsets  $\alpha$ . The resulting matrices are then used as the new  $Z^0$ ,  $M^0$  and  $S^0$  in the next PRSM call, i.e., those are used as a warm start.

We determine the order of variables for branching as follows. First, we consider for  $b \in \{-n, \dots, n\} \setminus \{0\}$ , the values

$$u_b = m_{\phi'} - p_{\phi'}^\emptyset + \langle \mathbf{I}, \widetilde{M} \rangle - \sum_{|b| \in \alpha} \widetilde{M}_{\alpha, \alpha}, \text{ for } \phi' = \mathbf{unitRes}(\phi, b), \quad (6.71)$$

similar to (6.56). Here,  $\widetilde{M}$  is the approximate solution to  $P_\phi$ . We remark that (6.71) can be quickly computed without explicitly performing the unit resolution. Observe that  $\langle \mathbf{I}, \widetilde{M} \rangle - \sum_{|b| \in \alpha} \widetilde{M}_{\alpha, \alpha}$  equals the trace of the new matrix  $M^0$ , which is used as warm start for  $P_{\phi'}$ . The value  $u_b$  is an estimate of the upper bound for the MAX-SAT problem corresponding to  $\phi'$ .

Second, we perform the rounding procedure to  $\widetilde{M}$  as described in Section 6.7.3. That is, we randomly generate a set  $\Lambda$  of vectors drawn uniformly at random on the unit sphere, and compute the corresponding rounded truth assignments  $x_\lambda \in \{\pm 1\}^{n_\phi}$ ,  $\lambda \in \Lambda$ , by (6.59). We update LB if a better truth assignment is found. Let  $\Lambda^* \subseteq \Lambda$  contain all the vectors  $\lambda$  that satisfy  $F_\phi(x_\lambda) = m_\phi - \text{LB}$ , and, assuming  $\Lambda^* \neq \emptyset$ , set  $\mathbf{v} := \frac{1}{|\Lambda^*|} \sum_{\lambda \in \Lambda^*} x_\lambda$ . Note that  $-1 \leq \mathbf{v}_i \leq 1 \forall i \in [n]$  with equality if and only if  $x_i$  is assigned the same truth value for all  $x_\lambda$ ,  $\lambda \in \Lambda^*$ . We define

$$B := \{b \in \mathbb{Z} : 0 < |b| \leq n_\phi, b \in \mathbb{Z}, \mathbf{v}_{|b|} = -\text{sgn}(b)\},$$

and explain its purpose by an example. If  $-3 \in B$ , then  $x_3$  is assigned **true** by all  $x_\lambda$ ,  $\lambda \in \Lambda^*$ . Heuristically, branching by setting  $x_3 = -1$  would then hopefully lead to low upper bounds in the resulting search tree. This is advantageous because low upper bounds lead to faster pruning. In case  $\Lambda^* = \emptyset$ , we set  $B = \{-n_\phi, \dots, n_\phi\} \setminus \{0\}$ .

Lastly, for all  $b \in B$ , we sort them in increasing order of  $u_b$ , see (6.71), and store this order in vector  $\sigma$ . Thus, the entries of  $\sigma$  satisfy

$$u_{\sigma_j} \leq u_{\sigma_{j+1}} \text{ and } \sigma_j \in B. \quad (6.72)$$

Vector  $\sigma$  determines the variable selection in the branching process of SOS-MS, which we describe in more detail in the sequel.

Consider a node in the SOS-MS search tree, in which we consider the proposition  $\phi$ . We initialize  $b_* := 0$ . For increasing values of  $j \in [b_{\max}]$ , where  $b_{\max} > 1$  is some integer (see (6.73)), we compute the SDP upper bound corresponding to  $\mathbf{unitRes}(\phi, \sigma_j)$ , denoted UB. In case  $\lfloor \text{UB} \rfloor \leq \text{LB}$ , we repeat this process with the next value of  $\sigma_j$ , and update  $b_* := b_* + 1$ . In case  $\lfloor \text{UB} \rfloor > \text{LB}$ , we terminate the process.

In case  $b_* > 0$ , we find that for all  $j \leq b_*$ , the propositions  $\mathbf{unitRes}(\phi, \sigma_j)$  cannot improve on LB. Thus, we may limit the search for better truth assignments to  $\mathbf{unitRes}(\phi, -\sigma_1, \dots, -\sigma_{b_*})$ . In case  $b_* = 0$ , we add both propositions  $\mathbf{unitRes}(\phi, \sigma_1)$

and  $\text{unitRes}(\phi, -\sigma_1)$  to the search tree, as we cannot exclude either one from attaining a value strictly greater than LB.

We take the previously mentioned  $b_{\max}$  as

$$\begin{aligned} b_{\max} &= \min \{ \max \{ 3; \lfloor 6 \text{GAP} + 1/2 \rfloor \}; 6 \}, \\ \text{for } \text{GAP} &= m_{\phi'} - p_{\phi'}^{\emptyset} + \langle \mathbf{I}, \widetilde{M} \rangle - \text{LB} - 1, \end{aligned} \tag{6.73}$$

where  $\widetilde{M}$  is an approximate solution to  $\mathbf{P}_{\phi}$ .

A pseudocode of SOS-MS is given by Algorithm 4. In particular, the branching process is described in Lines 15 to 20. Note the two PRSM calls in Line 8 and 16. In Line 8, the main purpose of the PRSM is to find an upper bound which equals the best known lower bound. When this does not occur, we use the approximate solutions as warm start for the PRSM call in Line 16. In Line 16, the purpose of the PRSM is to prune the node corresponding to  $\phi'$ . We use the LOBPCG algorithm [168] to efficiently approximate  $\lambda_{\min}(M^k)$ , which allows us to compute approximate upper bounds during the PRSM iterates, see (6.56) and (6.57). In case the approximate upper bound indicates that the node can be pruned, we recompute  $\lambda_{\min}(M^k)$  with the more accurate MATLAB `eig` function.

The algorithm can then stop iterating as soon as the condition in Line 17 is satisfied, or when it is clear that this condition cannot be satisfied in reasonable time.

**Remark 6.9.** Most of the running time of SOS-MS is spent on computing projections of matrices onto  $\mathcal{S}_+$ , as required by the PRSM. We found that computing the full eigenspectra of the matrices to be projected (in single-precision, rather than standard double-precision) using MATLAB's `eig` command was the fastest method of computing  $\mathcal{P}_{\mathcal{S}_+}(\cdot)$ , even though only the positive eigenpairs, or negative eigenpairs are required. For a variant of the PRSM, the authors of [266] propose using the LOBPCG algorithm [168] to compute only positive/negative eigenpairs (when this number is deemed small enough) of matrices to be projected. We were unable to obtain a speedup over `eig` through this method in the PRSM framework.  $\triangle$

### 6.9.1 Parsing sum of squares programs

We cover here the problem of initializing an SOS semidefinite program. Methods for achieving this are built in most SOS packages, such as SOSTOOLS [245] and GlobtiPoly [139]. In our application, we are interested in SOS modulo a vanishing ideal, which is not natively implemented in most SOS software, but rather, by restriction of the support set of the variables to a semialgebraic set (such as  $x^2 = 1$ ), which incurs additional variables in the SDP.

For most SDP applications, it is implicitly assumed that the program parameters are either already given, or require a negligible time to compute, in comparison to the time required for solving the resulting SDP. For most SOS programs however, this is decidedly not the case. The authors of SOSTOOLS [245], a third-party MATLAB package for formulating and solving SOS programs, confirm this observation. In chapter one of the user's manual to SOSTOOLS [245], it is stated that defining the semidefinite program, rather than solving it, is often the limiting factor for tractable

**Algorithm 4:** SOS-SDP MAX-SAT solver

---

```

1 Input: A MAX- $k$ -SAT instance  $\phi$  ( $k \leq 3$ ), on  $n_\phi$  variables and  $m_\phi$  clauses.
2 Output: Optimal truth assignment  $x \in \{\pm 1\}^n$ .
3 Set  $\text{UB} := m_\phi$ .
4 Use the CCLS algorithm on  $\phi$  to obtain a value for current best lower bound
   LB and corresponding truth assignment  $x \in \{\pm 1\}^n$ .
5 Initialize the stack  $Q := (\phi, \text{UB})$ .
6 while  $Q \neq \emptyset$  do
7   Take  $(\phi, \text{UB})$  as the first element of  $Q$ .
8   Use the PRSM, see (6.51), to obtain an approximate solution  $\widetilde{M}$ , see
   (6.57), to  $P_\phi$ .
9   Apply the rounding procedure from Sect. 6.7.3 to  $\widetilde{M}$  and obtain a lower
   bound LB. Update LB and  $x$  if a better truth assignment has been found.
10  Set  $\text{UB} := \min\{\text{UB}, \lfloor m_\phi - p_\phi^\emptyset + \langle \mathbf{I}, \widetilde{M} \rangle \rfloor\}$ , see (6.56).
11  Remove  $(\phi, \text{UB})$  from the stack  $Q$ .
12  if  $\text{LB} \geq \text{UB}$  then
13    /* The current node  $(\phi, \text{UB})$  can be pruned, so return to Line
14    6, and check if  $Q$  is empty. */
15    continue
16  Determine  $\sigma$ , see (6.72), and  $b_{\max}$ , see (6.73). Set  $b_* := 0$ .
17  for  $j = 1$  to  $b_{\max}$  do
18    Set  $\phi' := \text{unitRes}(\phi, \sigma_j)$ , use the PRSM to obtain an approximate
   solution  $\widetilde{M}$  to  $P_{\phi'}$ .
19    /* Use as warm start the matrices obtained from  $P_\phi$  in Line
20    8. */
21    if  $\text{LB} \geq \lfloor m_{\phi'} - p_{\phi'}^\emptyset + \langle \mathbf{I}, \widetilde{M} \rangle \rfloor$  then
22       $b_* := b_* + 1$ .
23    else
24      break
25  if  $b_* > 0$  then
26    Set  $\phi := \text{unitRes}(\phi, -\sigma_1, -\sigma_2, \dots, -\sigma_{b_*})$ , and  $Q := Q \cup (\phi, \text{UB})$ .
27    /*  $Q$  is nonempty, so return to Line 6. */
28  else
29    Set  $\phi_1 := \text{unitRes}(\phi, \sigma_1)$ ,  $\phi_2 := \text{unitRes}(\phi, -\sigma_1)$  and
    $Q := Q \cup \{(\phi_1, \text{UB}), (\phi_2, \text{UB})\}$ .
30    /*  $Q$  is nonempty, so return to Line 6. */
31 return Optimal truth assignment  $x$ .

```

---

problem size. In accordance with [245], we thus consider the problem of *parsing* an SOS program as defining it by its program parameters. This definition of parsing includes the problem of choosing the monomial basis  $\mathbf{x}$  such that the given polynomial can be accurately described. Many theoretical results can guide the choice of  $\mathbf{x}$ , such as the Newton polytope [290], see also [264], or facial reduction [252]. For our purposes however, choosing  $\mathbf{x}$  can be done with a simple fixed procedure, see e.g., (6.27). We therefore consider parsing as only the purely numerical problem of finding, rather

than choosing,  $\mathbf{x}$ .

In this section, we provide a short overview of our parsing method. We exploit the fact that our variables are  $\{\pm 1\}^n$ , which allows us to achieve fast parsing times, compared to general purpose SOS software. For example, due to the properties of computation modulo  $\mathcal{I}$ , see (6.21), monomials can be stored in two ways: either we store some  $\alpha \subseteq [n]$ , corresponding to  $x^\alpha$  as in (6.5), (*subset format*) or we store monomials as a vector  $\mathbf{v} \in \{0, 1\}^n$ , corresponding to  $x^\mathbf{v} = x_1^{v_1} \dots x_n^{v_n}$  (*vector format*). The vectors  $\mathbf{v}$  can be saved as boolean vectors. It is trivial to switch between these two formats, which is what we use in our parsing algorithm: we implement each step using the best suited format. Also note that for monomial basis  $\mathbf{x}$  given by (6.27), monomials in  $\mathbf{X} \equiv \mathbf{xx}^\top \pmod{\mathcal{I}}$ , see (6.26), will have degree at most 4, which ensures that the subset format requires little storage.

Let  $\phi$  be the considered proposition, on  $n$  variables and  $m$  clauses. Initially, to compute  $\mathbf{x}$  according to (6.27), we consider the unique clauses  $C_j$  of  $\phi$ ,  $j \in [m]$ . Recall that we define a clause  $C_j$  as a subset of  $[n]$ , see Section 6.1. To compute the monomials in  $\mathbf{X}$ , we need to compute the cross products of all monomials in  $\mathbf{x}$ . This is best done in vector format, by using the entrywise *exclusive or* operation on boolean vectors, denoted  $\oplus$ . That is,  $x^\mathbf{v}x^\mathbf{u} \equiv x^{\mathbf{v} \oplus \mathbf{u}} \pmod{\mathcal{I}}$ , which is computationally cheaper than the symmetric difference operator as in (6.26).

Next, to construct the sets of indices  $\mathbf{x}^\gamma$ , as in (6.30), we need to find the sets of equal monomials in the strictly upper triangular part of  $\mathbf{X}$ . We first divide the monomials in the upper triangular part of  $\mathbf{X}$  in four groups, based on their degree. The degrees are trivially computed in vector format as  $\mathbf{1}_n^\top \mathbf{v} \in \{1, 2, 3, 4\}$ . For each group, we switch to subset format. In particular, we store the monomial  $x^\gamma$ , of degree  $d \in \{1, 2, 3, 4\}$ , as a sorted vector  $g \in \mathbb{N}^d$ . Given these sorted vectors  $g$ , we use the MATLAB `unique` function to find the subgroups of equal monomials. We save their corresponding indices in  $\mathbf{X}$  to construct the sets  $\mathbf{x}^\gamma$ .

We compute the coefficients  $p_\phi^\gamma$ , see (6.24), iteratively per clause. If  $\phi$  induces a MAX- $k$ -SAT problem, we create  $k + 1$  matrices  $A_s$ ,  $s \in \{0, 1, \dots, k\}$ , where matrix  $A_s$  is an  $s$ -dimensional matrix. Then, for  $\gamma = \{\gamma_1, \dots, \gamma_s\}$ , we store  $p_\phi^\gamma$  at position  $(A_s)_{\gamma_1, \dots, \gamma_s}$ . Note that matrix  $A_0$  is a number storing  $p_\phi^\emptyset$ , see (6.25). For MAX-3-SAT instances, matrices  $A_1$  and  $A_2$  will be generally be dense and matrix  $A_3$  will generally be sparse. We have also tested the recently developed `dpvar` structure of SOSTOOLS [149] to compute the  $p_\phi^\gamma$  symbolically, but found that it compared unfavourably to our procedure, in terms of computation time.

The last step is to match the coefficients  $p_\phi^\gamma$  to the monomial  $x^\gamma$  of  $\mathbf{X}$ . Clearly, we need only to consider those coefficients  $p_\phi^\gamma$  for which  $p_\phi^\gamma \neq 0$ . Thus, for some nonzero  $p_\phi^\gamma$  stored in  $A_s$ , we know that  $|\gamma| = s$ . As the monomials in  $\mathbf{X}$  have already been divided into groups based on their degree, we search only the monomials of degree  $s$ , to find  $x^\gamma$ . In case we use the  $SOS_p$  basis, then by Lemma 6.2, we need only check those monomials  $x^\alpha$  for which  $|\mathbf{x}^\alpha| > 2$ , see (6.30). Now to find  $x^\gamma$  in this subgroup of monomials (of which one equals  $x^\gamma$ ), we use the vector format.

By exploiting properties of the ideal  $\mathcal{I}$ , see (6.21), and  $SOS_p$ , we are able to obtain high parsing speeds. For example, instance `s3v70c1500-1.cnf` (70 variables and 1500

clauses of length 3) from the MSE-2016<sup>3</sup>, induces an  $SOS_p$  basis of size 2107. Matrix  $\mathbf{X}$ , see (6.26), then contains 4,439,449 monomials. Our algorithm parses this basis in (approximately) 1 second. This completes the summary of our parsing algorithm. For more details, interested readers are referred to the code available on [github](#)<sup>4</sup>.

## 6.10 Numerical results

In this section, we test SOS-MS, described in Section 6.9, on MAX-3-SAT, weighted partial MAX-2-SAT, and weighted MAX-3-SAT instances from the MSE-2016. We also compute SDP bounds on some partial MAX-3-SAT instances from the same source. We choose the year 2016, because later years of the MSE offer no MAX-2-SAT or MAX-3-SAT instances. In particular, the instances in this section are taken from the MSE-2016<sup>5</sup> random track.

Experiments are carried out on a 16 GB RAM laptop, with an Intel i7-1165G7 CPU (2.8 GHz) and four cores, running Windows 10 Enterprise. We set the PRSM parameters, see (6.51), as

$$(\nu_1, \nu_2, \rho) = \left( \frac{1}{2}, \frac{9}{10} \frac{1 + \sqrt{5}}{2}, \frac{2s}{5} \right),$$

where  $s$  is the order of the matrix variable. Our MATLAB implementation of the PRSM is available at [https://github.com/LMSinjorgo/SOS-SDP\\_MAXSAT](https://github.com/LMSinjorgo/SOS-SDP_MAXSAT).

### 6.10.1 MAX-3-SAT problems

We compare MAX-3-SAT upper bounds obtained by solving  $P_\phi$  on Page 158, for various instances and monomial bases. We also demonstrate the performance of SOS-MS on MAX-3-SAT instances.

Table 6.1 presents a comparison of the bounds obtained by solving the SOS program  $P_\phi$  using the  $SOS_p$  basis (6.27) and the  $SOS_p^{\mathbf{Q}}$  bases (6.28) where  $\mathbf{Q} \in \{40, 50, 60, 70, 110\}$ . We use for several instances from the MSE-2016 MAX-3-SAT category on 70, 90 and 110 variables. The first column reports the name of the corresponding instance, on  $v$  variables and  $c$  clauses. Column LB provides the best found lower bound, obtained by running the CCLS algorithm [210] for five seconds. Column UB reports the computed upper bounds. Column **Iter.** gives the number of PRSM iterations, divided by  $10^2$ . For each instance and monomial basis, we run 200 iterations. Then, if the observed value of UB satisfies  $UB \leq LB + 1.5$ , we perform 200 additional iterations. We stop early if  $\lfloor UB \rfloor = LB$ . The results show the strength of the  $SOS_p$  basis for instances with with 70 variables. In particular, that basis is sufficiently large for closing gaps of several instances. The results also show that the  $SOS_p^{\mathbf{Q}}$  basis can be used to further improve bounds for instances with 70, 90 and 110 variables.

In Table 6.2, we present more details on the best upper bounds attained on the same instances as in Table 6.1. The columns of Table 6.2 follow the same definitions

<sup>3</sup>All instances are available at <http://www.maxsat.udl.cat/16/benchmarks/index.html>.

<sup>4</sup>Code available at [https://github.com/LMSinjorgo/SOS-SDP\\_MAXSAT](https://github.com/LMSinjorgo/SOS-SDP_MAXSAT).

<sup>5</sup>For more information on the MSE-2016, see <http://www.maxsat.udl.cat/16/index.html>.

as the previous table. Additionally, column  $\mathbf{Q}$  relates to the  $SOS_p^{\mathbf{Q}}$  basis used and column  $|\mathbf{x}|$  reports the number of monomials in that basis (equivalently, the order of the matrix variable of program  $P_\phi$ ). Here, we refer to  $SOS_p$  as  $SOS_p^0$ . Column  $\mathbf{T}$ . (s) reports the computation time in seconds. The results show that we closed the optimality gap for all instances with 70 and 90 variables in less than 9 minutes. Table 6.2 also shows that the computational time increases w.r.t. the size of the basis. Furthermore, Table 6.2 shows that uniform random MAX-SAT instances on the same number of variables and clauses can differ in difficulty to solve. For example, instances `s3v70c800-1.cnf` and `s3v70c800-3.cnf` have the same number of variables and clauses. However, proving optimality of the lower bound of the former requires almost three times the computation time as for the latter.

In Figure 6.1, we show the performance of SOS-MS, using the  $SOS_p^{50}$  basis, on all the MAX-3-SAT instances on 70 variables from the MSE-2016. For detailed running times see Appendix A.2 on Page 191. We compare the running times of SOS-MS with the corresponding running times of the best results of the MSE-2016. That is, for each instance, we compare SOS-MS with the participant of the MSE-2016 that was able to solve that specific instance in the least time. All solvers at the MSE-2016 were tested on an Intel Xeon E5-2620 processor with 2.0 GHz and 3.5 GB RAM<sup>6</sup>. It is hard to compare the running times of algorithms on different machines, since it depends on many factors such as the processor, RAM, the operating system, et cetera. We choose to consider the difference in clock speeds, i.e., 2.8 GHz vs. 2.0 GHz, as well as wall time. Therefore, in Figure 6.1 and part of Table A.1 in the appendix on Page 191, we have multiplied all the original running times of SOS-MS with 1.4. Additionally, as the solvers in the MSE-2016 were given a maximum time of 30 minutes per instance, we provided SOS-MS with a maximum of  $30/1.4 (\approx 21.4)$  minutes.

Under these constraints, SOS-MS is able to solve all 45 MAX-3-SAT instances on 70 variables. The participants from the MSE-2016 could solve at most 42 instances. Specifically, they were unable to solve `s3v70c1500-1`, `s3v70c1500-4` and `s3v70c1500-5`. For instances with a lower number of clauses (around 800) however, the best times per instance of the MSE-2016 are much lower than for SOS-MS. SOS-MS takes on average 309.10 seconds per solved instance, compared to 367.7 seconds per solved instance for the best MSE-2016 solvers. We have also tested SOS-MS using the  $SOS_p$  basis, on the same instances. With this different basis, SOS-MS was also able to solve all 45 instances, taking on average 316.7 seconds per instance.

We also investigate the performance of SOS-MS on MAX-3-SAT problems with 80 booleans, from the MSE-2016 database. These instances were not tested in the MSE-2016, and thus, it is not known what the best solver per instance is. On these instances, we compare SOS-MS to the CCLS2akms MAX-SAT algorithm<sup>7</sup>. This algorithm first runs CCLS [210] to find a good starting lower bound for the MAX-SAT solution. It then passes this lower bound to the akmaxsat algorithm [170], which solves the instance to optimality. Out of all the publicly available solvers, CCLS2akms

---

<sup>6</sup>The full specifications of this machine are available at <http://www.maxsat.udl.cat/16/machinespecifications/index.html>.

<sup>7</sup>The CCLS2akms algorithm is available under the name CCLS\_to\_akmaxsat at <http://www.maxsat.udl.cat/16/solvers/index.html>.

placed highest in the random MAX-SAT category of MSE-2016<sup>8</sup>.

Results for the MAX-3-SAT instances on 80 variables from MSE-2016 are given in Figure 6.2. The running times per tested MAX-3-SAT instance are provided in Appendix A.2, Table A.1. Both SOS-MS and CCLS2akms are provided a maximum of 30 minutes per instance and are tested on the same hardware (our 16 GB RAM laptop), and thus not scaled. For SOS-MS, we used the  $SOS_p$  basis, as  $SOS_p^Q$  for  $Q = 40$  and  $Q = 55$  provided worse results. CCLS2akms is able to solve 40 of the 48 instances within the time limit, while SOS-MS solves 43. SOS-MS is however slower: it requires on average 772.53 seconds per solved instance, compared to 370.48 per instance. Again, SOS-MS performs well for instances with a large number of clauses, but requires more time for those with a low number of clauses.

### 6.10.2 Weighted partial MAX-2-SAT problems

We investigate the performance of our solver SOS-MS on (weighted) partial MAX-2-SAT instances from MSE-2016. For this purpose, we adjust SOS-MS by using the theory outlined in Section 6.8.

In particular, we perform a B&B search, using (6.65) to compute upper bounds. However, we first preprocess an instance by using the SAT resolution rule (6.68) on the hard clauses until we find all implied hard clauses of length two or less. Note that this preprocessing may result in improved upper bounds, as described in Section 6.8.2. If hard unit clauses are found this way, we perform unit resolution and continue with the reduced problem. As initial lower bound for our solver, we take the best known lower bound reported in MSE-2016.

Note that in case of a (weighted) partial MAX-SAT instance, truth values assigned during branching might create hard unit clauses, which leads to more forced truth assignments. Additionally, if a node contains few unassigned variables, determining good upper bounds can be done with a small monomial basis, such as  $SOS_s^\theta$  (6.29) for small values of  $\theta$ . Let us describe the choice of  $\theta$  through the B&B tree. We initialize  $\theta_{\text{start}} = 0$ . At a node in which we compute bounds, we compute an upper bound UB to the (weighted) partial MAX-SAT solution by first using the basis  $SOS_s^{\theta_{\text{start}}}$ . If  $[\text{UB}] \leq \text{LB}$ , we prune the current node. If not, we consider the value  $\text{GAP} = \text{UB} - \text{LB} > 0$ . When  $\text{GAP} < 10$ , we set  $\theta = 0.5$  and recompute a stronger upper bound. For  $\text{GAP} \geq 10$ , we recompute an upper bound with  $\theta = 1$  instead. In both cases, we use the PRSM variables corresponding to  $\theta_{\text{start}}$  as warm starts for the next PRSM. If the upper bound obtained using  $\theta \in \{0.5, 1\}$  is not equal LB, we set  $\theta_{\text{start}} = 0.1$  for the remainder of the algorithm, and continue with the B&B.

We determine our branching variable in the following way. Let  $n', n' \leq n$ , be the number of remaining variables at the current node. We consider the five truth assignments that create the largest number of hard unit clauses. For each of these five truth assignments  $\sigma_i, i \in [5], \sigma_i \in \{-n', \dots, n'\} \setminus \{0\}$ , we compute  $\phi_i = \text{unitRes}(\phi, \sigma_i)$ , see Section 6.9. Then, we select the truth assignment  $\sigma_i$  for which the polynomial  $\phi_i$  has the largest constant term, see (6.25). The branching variable is then given by  $|\sigma_i|$ , and the two children nodes correspond to the propositions  $\text{unitRes}(\phi, \sigma_i)$  and

<sup>8</sup>The MSE-2016 results are available at <http://www.maxsat.udl.cat/16/solvers/index.html>.

$\text{unitRes}(\phi, -\sigma_i)$ . This branching rule aims to create nodes with few remaining variables, due to the presence of many hard unit clauses. This allows for setting  $\theta_{\text{start}}$  to small values, while still providing strong bounds, see also Appendix A.4.

The computation of SOS-SDP based upper bounds is expensive, compared to bounding methods used in other MAX-SAT solvers. Therefore, we only compute bounds at selected nodes (see also [310]). A method for determining at which nodes to compute bounds is described in detail in Appendix A.3 on Page 193. Appendix A.3 also provides a pseudocode of SOS-MS on weighted partial MAX-2-SAT problems.

We test the described procedures on the 60 unweighted partial MAX-2-SAT instances, and the 90 weighted partial MAX-2-SAT instances from the MSE-2016, setting a maximum time of 30 minutes per instance. Each instance contains 150 variables and 150 hard clauses. The number of total clauses (both soft and hard) ranges from 1000 to 5000, and all of them have length two. In the weighted variant, soft clause weights range from 1 up to and including 10. Tables 6.3 and 6.4 report the running times per instance (rounded to the nearest second), for unweighted and weighted partial MAX-2-SAT, respectively. Here we provide original runtimes, thus not multiplied by some factor. A ‘-’ value indicates a time-out of 30 minutes. Variable  $m$  denotes the number of total clauses. The **Instance** row corresponds to the instance file name, as taken from the MSE-2016. For example,  $m = 2500$  and **Instance** = 1 refer to `file_rpms_wcnf_L2_V150_C2500_H150_1.wcnf` in Table 6.3, and `file_rwpms_wcnf_L2_V150_C2500_H150_1.wcnf` in Table 6.4.

The table shows that we are able to solve many instances within the 30-minute time limit. This shows the strength of SDP applied also to the (weighted) partial MAX-SAT problem. Since we are first to solve the (weighted) partial MAX-SAT problem by using SDP approaches, this chapter opens new perspectives on solving variants of the MAX-SAT problem.

In Appendix A.4 on Page 195 we provide the full search tree for three partial MAX-2-SAT instances.

### 6.10.3 Partial MAX-3-SAT problems

We show the quality of the SDP bounds for the partial MAX-3-SAT problem, based on 10 instances from the MSE-2016.

We compute an upper bound for each instance  $\phi$  (on  $n$  variables, having hard clauses  $C_p^{\mathbf{H}}$ ) in the following way. We first perform SAT resolution (6.68) on the hard clauses to find all implied hard clauses of length 4 or less. Then, for  $\mathbf{Q} \in [n]$ , we consider the  $\mathbf{Q}$  variables that appear in the largest number of (soft and hard) clauses. Let  $V \subseteq [n]$  be the subset indicating those variables. We construct additional hard clauses of the form  $C_p^{\mathbf{H}} \vee x_i$ , for all  $i \in V$  and  $C_p^{\mathbf{H}} \subseteq V$ , with  $|C_p^{\mathbf{H}}| \leq 3$ . Note that these additional hard clauses do not change the set of satisfying assignments. On the instance generated in this way, we compute an upper bound using the  $SOS_p^{\mathbf{Q}}$  basis, see (6.28). Note that, as the newly generated clauses  $C_p^{\mathbf{H}} \vee x_i$  are contained in  $V$ , they create no additional monomials in the  $SOS_p^{\mathbf{Q}}$ . This ensures that the size of the matrix variable remains manageable. Moreover, these additional hard clauses strengthen the bound, see Section 6.8.2.

We perform this procedure for each instance and  $\mathbf{Q} \in \{70, 75, 80, 85, 90\}$ , for a

time limit of 30 minutes. These values of  $\mathbf{Q}$  are chosen with the goal of computing the tightest bound at the 30-minute mark. We report results in Table 6.6. Column `id` reports the instance identifier, according to the naming scheme

`file_rpms_wcnf_L3_V100_C600_H100_[id].wcnf.`

Each instance has 600 clauses, of which 100 are hard, all of length three, on 100 variables. Column `LB` reports the optimal lower bound, as verified by solvers in the MSE-2016. Column  $\mathbf{Q}$  refers to the  $SOS_p^{\mathbf{Q}}$  basis used, of which the number of monomials is reported in column `|x|`. Column `|CH|` reports the number of hard clauses used. The next 6 columns (`GAP at X minutes`) report the value of the `GAP` (i.e., `UB - LB`) at different time points. To compute the values of `UB`, we compute the smallest eigenvalue of  $M$  using the LOBPCG algorithm [168], see (6.57). Lastly, as a measure of convergence, the final column reports the (absolute) value of the smallest eigenvalue of the matrix variable at the final iteration, multiplied by the size of this matrix. This column thus reports the difference in trace between  $\widetilde{M}$  and  $M$ , see (6.57).

Considering the bound at 30 minutes, the  $SOS_p^{85}$  basis performs best. A larger basis is unable to converge in 30 minutes, while smaller bases result in weaker bounds. Since differences in bounds for different  $\mathbf{Q}$  are small, smaller bases might be more useful in combination with a B&B scheme.

#### 6.10.4 Weighted MAX-3-SAT problems

Lastly, we test SOS-MS on some weighted MAX-3-SAT instances. We consider 10 weighted MAX-3-SAT instances from the MSE-2016. Each instance contains 70 variables, and either 1400 or 1500 weighted soft clauses. The clause weights range between 1 and 10. There are no hard clauses.

For the weighted MAX-3-SAT problem, we compare the running times of SOS-MS with CCLS2akms, on the same hardware. For SOS-MS, we attempted multiple monomial bases, and found that the  $SOS_p$  basis required the least time to solve the weighted MAX-3-SAT instances.

The running times per instance are reported in Table 6.5. Column `m` reports the number of clauses, and `Inst.` reports the instance, according to the scheme `s3v70c[m]-[Inst.].wcnf`. SOS-MS is able to solve three instances in less time than CCLS2akms and can solve 9 of the 10 instances in less than 30 minutes. This demonstrates that SOS-MS is well suited for solving weighted MAX-3-SAT instances with a large number of clauses.

## 6.11 Conclusions

In this chapter we have considered SOS optimization for solving the MAX-SAT and weighted partial MAX-SAT problems. We design an SOS-SDP based exact MAX-SAT solver, called SOS-MS. Our solver is competitive with the best-known solvers on solving various (weighted partial) MAX-SAT instances. We are also first to compute SDP bounds for the weighted partial MAX-SAT problem.

In Section 6.3 we propose a family of semidefinite feasibility problems  $R_{\mathcal{B}}(\phi)$  and show that one member of this family provides the rank two guarantee, see Theorem 6.1. That is, the existence of a feasible rank two matrix implies satisfiability of the corresponding SAT instance. In Section 6.4, we outline the SOS approach to the MAX-SAT problem, due to van Maaren et al. [299] and propose new bases. We introduce the  $SOS_s^\theta$  and  $SOS_p^{\mathbf{Q}}$  bases, see (6.29), and provide several theoretical results related to these bases, see Lemmas 6.2 and 6.3. Clearly, the strength of the SOS-SDP based relaxations and the required time to compute them depend on the chosen monomial basis. The SOS-SDP relaxation for the MAX-SAT problem is denoted by  $P_\phi$ . We consider MAX-SAT resolution in Section 6.5 and show that resolution might not be beneficial for the SOS approach applied to the MAX-SAT problem.

In Section 6.6, we elegantly show a connection between the SOS approach to the MAX-SAT problem and the family of semidefinite feasibility problems  $R_{\mathcal{B}}(\phi)$ . This is done by deriving the dual problem to  $P_\phi$ , see Theorem 6.6. In Section 6.7, we propose the PRSM for solving  $P_\phi$ . We show that the PRSM is well suited for exploiting the structure of  $P_\phi$ , in particular, the unit constraints, see (6.33). We thus provide an affirmative answer to the key question posed by van Maaren et al. [299]: “*Whether SDP software can be developed dealing with unit constraints efficiently?*”.

We extend the SOS approach for the MAX-SAT problem to the weighted partial MAX-SAT problem in Section 6.8. Here, the variables are restricted to satisfy a set of hard clauses. We show that such hard clauses can be incorporated in the SOS program  $P_\phi$  by adding scalar variables. We show in Section 6.8.1 that the resulting program (6.64) is also well suited for the PRSM.

In Section 6.9, we provide implementation details of our SOS-SDP based exact MAX-SAT solver, whose pseudocode is given in Algorithm 4. SOS-MS is a B&B algorithm and has two crucial components. The first one is the use of warm starts for program  $P_\phi$ , in order to quickly obtain strong bounds. The second one is its ability to quickly parse  $P_\phi$ , as outlined in Section 6.9.1. Our algorithm parses a basis that contains 4,439,449 monomials in (approximately) one second (!).

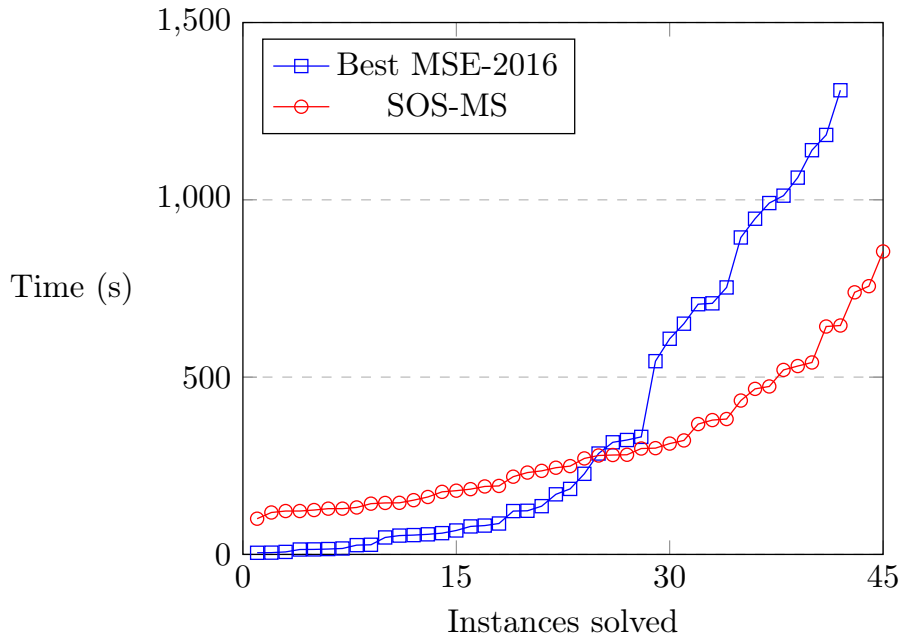
In Section 6.10 we provide extensive numerical results that verify efficiency of our exact solver SOS-MS and quality of SOS upper bounds. We show that SOS-MS can solve a variety of MAX-SAT instances in reasonable time, while solving some instances faster than the best solvers in the MSE-2016. We show that the  $SOS_p^{\mathbf{Q}}$  bases (6.28) are able to prove optimality of some MAX-SAT instances, and that the parameter  $\mathbf{Q}$  provides the option to adjust the trade-off between quality of the bounds and computation time. We also test our B&B algorithm for (weighted) partial MAX-SAT instances in Sections 6.10.2 and 6.10.4. Our solver is able to solve many (weighted) partial MAX-SAT instances in a reasonable time.

This chapter has demonstrated the strong performance of SOS-MS on (weighted partial) MAX-SAT instances from the MSE random track. In the future, we hope to also solve instances with SOS-MS from the so-called *industrial* and *crafted* tracks. These tracks currently impose two challenges on SOS-MS. Firstly, these instances induce prohibitively large  $SOS_p$  bases, which hinders the computation of strong bounds. To solve this, we require a more sophisticated method for choosing a smaller, manageable, basis, like  $SOS_s^\theta$ . Secondly, these instances can possess clauses of length  $k$ , where  $k \geq 4$ . This is problematic in the current settings, since  $F_\phi$ , see (6.19), is a

$k$ th degree polynomial, which requires a large basis to be represented. One possible way to overcome these challenges is through exploiting the structure present in these instances. For example, function  $F_\phi$  might have few nonzero coefficients, which allows for finding *SOS* decompositions with small monomial bases, using methods proposed in [307], see also [5].

Instance	$SOS_p$		$SOS_p^{40}$		$SOS_p^{50}$		$SOS_p^{60}$		$SOS_p^{70}$		$SOS_p^{110}$	
	LB	UB	Iter.	UB	Iter.	UB	Iter.	UB	Iter.	UB	Iter.	UB
s3v70c800-1	769	771.29	2	770.79	2	770.67	2	769.99	3.9			
s3v70c800-3	770	770.996	3									
s3v70c800-4	772	772.99	2.8									
s3v70c900-4	861	863.11	2	862.63	2	861.99	3.3					
s3v70c1000-1	953	954.89	2	954.70	2	954.65	2	953.999	3.5			
s3v70c1000-2	957	957.99	2.7									
s3v70c1000-5	958	958.996	2.2									
s3v70c1100-4	1048	1049.03	4	1048.997	3.2							
s3v70c1500-2	1411	1411.998	2.8									
s3v90c900-5	875	877.44	2	875.99	2.9							
s3v90c900-7	873	877.16	2	875.56	2	875.05	2	874.62	2	873.995	2.7	
s3v110c1000-7	969	984.01	2	980.93	2	979.77	2	978.63	2	977.61	2	974.49
s3v110c1100-10	1064	1076.81	2	1074.15	2	1073.03	2	1071.98	2	1071.01	2	1068.32

Table 6.1: Comparison of the MAX-3-SAT bounds attained by different monomial bases.

Figure 6.1: SOS-MS on 70 variable MAX-3-SAT (basis  $SOS_p^{50}$ )

Instance	LB	UB	T. (s)	Q	$ x $	Iter.
s3v70c800-1	769	769.99	243.0	60	2181	3.9
s3v70c800-3	770	770.996	85.0	0	1603	3.0
s3v70c800-4	772	772.99	80.7	0	1588	2.8
s3v70c900-4	861	861.99	168.4	50	2022	3.3
s3v70c1000-1	953	953.999	236.6	60	2244	3.5
s3v70c1000-2	957	957.99	102.4	0	1810	2.7
s3v70c1000-5	958	958.996	83.6	0	1798	2.2
s3v70c1100-4	1048	1048.997	164.4	40	2014	3.2
s3v70c1500-2	1411	1411.998	165.7	0	2130	2.8
s3v90c900-5	875	875.99	218.4	40	2366	2.9
s3v90c900-7	873	873.995	519.9	70	3185	2.7
s3v110c1000-7	969	974.49	3089.3	110	6106	2.0
s3v110c1100-10	1064	1068.32	3232.5	110	6106	2.0

Table 6.2: Best upper bounds for the MAX-3-SAT problem per instance

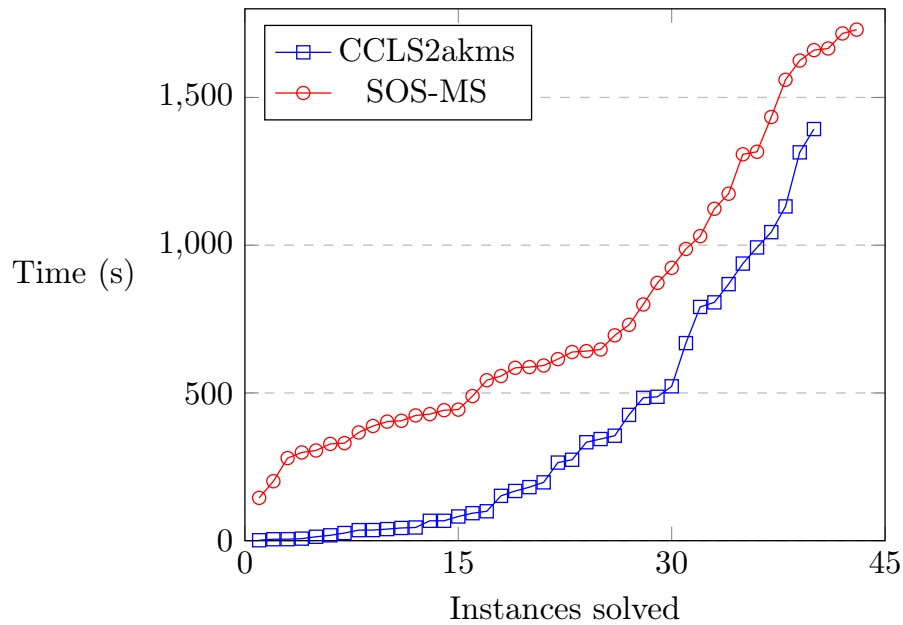
id	Q	$ x $	$ C^H $	GAP at $X$ minutes						$ x \lambda_{\min}$ ( $\times 10^{-2}$ )
				5	10	15	20	25	30	
70	3288	5188	4.65	3.55	3.16	2.95	2.82	2.72	4.01	
75	3507	6124	4.81	3.51	3.07	2.82	2.67	2.56	5.82	
0	80	3754	7140	5.35	3.61	3.03	2.72	2.53	2.40	9.64
85	4035	7908	6.31	4.08	3.20	2.78	2.52	2.34	18.54	
90	4340	8629	7.54	4.76	3.70	3.14	2.78	2.53	41.27	

id	Q	x	C <sup>H</sup>	GAP at X minutes						x λ <sub>min</sub> (×10 <sup>-2</sup> )
				5	10	15	20	25	30	
1	70	3334	4412	4.37	3.48	3.16	2.97	2.84	2.75	2.80
	75	3548	5509	5.03	3.58	3.13	2.88	2.72	2.60	4.49
	80	3780	6331	5.56	3.66	3.11	2.81	2.63	2.50	6.95
	85	4054	7377	5.84	3.60	2.97	2.65	2.44	2.30	9.41
	90	4352	8329	7.77	4.44	3.46	2.97	2.67	2.46	25.62
2	70	3324	5067	4.09	2.99	2.58	2.35	2.20	2.10	5.29
	75	3536	5993	4.47	3.05	2.56	2.29	2.11	1.98	9.78
	80	3776	7076	4.92	3.18	2.60	2.27	2.06	1.91	15.80
	85	4046	7839	5.05	3.25	2.51	2.16	1.93	1.77	23.72
	90	4346	8729	7.02	4.12	3.18	2.67	2.34	2.10	52.53
3	70	3296	4964	5.33	4.42	4.04	3.82	3.68	3.58	1.58
	75	3528	5741	5.33	4.26	3.84	3.60	3.44	3.33	2.24
	80	3773	6501	5.79	4.26	3.76	3.49	3.31	3.18	3.47
	85	4038	7475	5.94	4.15	3.62	3.33	3.14	3.00	4.56
	90	4343	8345	8.18	4.92	4.02	3.58	3.31	3.12	10.29
4	70	3304	4832	4.10	3.14	2.74	2.52	2.38	2.28	3.24
	75	3529	5815	4.54	3.17	2.70	2.44	2.26	2.13	5.41
	80	3775	6811	4.62	3.06	2.55	2.27	2.08	1.95	7.06
	85	4052	7641	5.23	3.17	2.55	2.21	1.99	1.84	11.19
	90	4352	8695	7.53	4.16	3.11	2.60	2.28	2.05	25.95
5	70	3287	5207	3.96	3.15	2.82	2.64	2.52	2.44	1.80
	75	3506	6071	4.06	3.15	2.74	2.52	2.37	2.27	2.82
	80	3760	6851	4.40	3.11	2.66	2.40	2.24	2.12	4.29
	85	4037	7927	4.99	3.29	2.76	2.46	2.26	2.12	7.77
	90	4347	8648	6.70	4.05	3.25	2.83	2.54	2.34	22.01
6	70	3298	4633	3.70	2.84	2.49	2.29	2.16	2.07	3.60
	75	3508	5745	3.94	2.82	2.42	2.19	2.03	1.93	5.45
	80	3744	6659	4.27	2.82	2.38	2.13	1.97	1.84	9.95
	85	4018	7643	4.87	3.11	2.38	2.06	1.86	1.71	15.91
	90	4332	8523	7.29	3.97	3.03	2.55	2.25	2.02	40.32
7	70	3345	4980	5.38	4.44	4.05	3.83	3.68	3.58	1.26
	75	3562	6117	5.48	4.42	3.97	3.72	3.55	3.43	1.75
	80	3803	7056	5.89	4.42	3.92	3.62	3.43	3.30	2.50
	85	4062	7901	5.88	4.25	3.73	3.43	3.23	3.09	3.02
	90	4359	8709	7.50	4.68	3.93	3.56	3.31	3.12	8.27
8	70	3258	5155	3.83	2.65	2.23	1.99	1.84	1.72	9.72
	75	3484	5803	4.09	2.65	2.17	1.90	1.72	1.59	15.76
	80	3735	6722	4.85	3.02	2.29	1.96	1.75	1.59	27.59
	85	4008	7547	5.11	3.08	2.37	1.90	1.66	1.50	39.49
	90	4324	8770	7.55	4.18	3.09	2.52	2.17	1.92	76.63

id	$\mathbf{Q}$	$ \mathbf{x} $	$ C^{\mathbf{H}} $	GAP at $\mathbf{X}$ minutes						$ \mathbf{x} \lambda_{\min}$ ( $\times 10^{-2}$ )
				5	10	15	20	25	30	
	70	3341	5206	5.33	4.39	4.00	3.78	3.63	3.53	1.93
	75	3569	5926	5.43	4.31	3.85	3.59	3.42	3.31	2.59
9	80	3811	7158	5.78	4.27	3.75	3.46	3.27	3.14	3.60
	85	4078	8254	5.92	4.17	3.63	3.32	3.11	2.97	4.55
	90	4367	9169	7.57	4.80	4.00	3.56	3.28	3.08	11.35

Table 6.6: Bounds for partial MAX-3-SAT instances

Figure 6.2: SOS-MS on 80 variable MAX-3-SAT (basis  $SOS_p$ )



		Instance									
		0	1	2	3	4	5	6	7	8	9
$m$	2500	59	177	237	536	75	45	400	361	320	61
	3000	327	329	96	244	103	554	675	86	339	221
	3500	322	223	112	134	188	620	23	469	144	252
	4000	85	5	107	127	347	662	762	320	-	289
	4500	679	334	592	251	732	470	145	223	318	135
	5000	159	116	663	1258	975	226	473	598	200	105

Table 6.3: Unweighted 150 variable partial MAX-2-SAT running times (seconds)

		Instance									
		0	1	2	3	4	5	6	7	8	9
$m$	1000	116	164	61	40	54	85	55	196	344	68
	1500	608	144	447	164	529	190	105	269	349	370
	2000	325	495	326	222	134	233	124	156	178	631
	2500	103	544	282	1029	318	315	575	926	619	118
	3000	-	1341	446	-	249	-	1220	422	1624	618
	3500	1667	1195	1022	450	1327	1351	130	771	196	229
	4000	91	5	208	1108	930	-	-	1048	-	338
	4500	-	-	1601	1220	-	732	518	1347	799	965
	5000	338	980	-	686	-	-	-	-	222	294

Table 6.4: Weighted 150 variable partial MAX-2-SAT running times (seconds)

$m$	Inst.	Running time (s)	
		SOS-MS	CCLS2akms
1400	1	438.71	322.62
	2	795.71	278.78
	3	696.16	752.95
	4	852.51	592.82
	5	1250.02	1054.16
1500	1	499.26	885.66
	2	653.51	1011.90
	3	399.28	292.60
	4	791.08	748.03
	5	> 1800.00	1323.78

Table 6.5: Weighted 70 variable MAX-3-SAT on  $m$  clauses

# A Supplementary data

## A.1 Scaled form of the ADMM

Consider the unscaled ADMM scheme (1.29) on Page 14. Define  $\tilde{Z}^\ell := \frac{1}{\rho}Z^\ell$  and  $\tilde{C} := \frac{1}{\rho}C$ . Note that (1.29) can be equivalently stated in terms of  $\tilde{Z}^\ell$  and  $\tilde{C}$  as:

$$\begin{aligned} X^{\ell+1} &= \mathcal{P}_{S_+^n}(Y^\ell + \tilde{Z}^\ell) \\ \tilde{Z}^{\ell+\frac{1}{2}} &= \tilde{Z}^\ell + \nu_1(Y^\ell - X^{\ell+1}) \\ Y^{\ell+1} &= \mathcal{P}_{\mathcal{F}}(X^{\ell+1} - \tilde{C} - \tilde{Z}^{\ell+\frac{1}{2}}) \\ \tilde{Z}^{\ell+1} &= \tilde{Z}^{\ell+\frac{1}{2}} + \nu_2(Y^{\ell+1} - X^{\ell+1}), \end{aligned} \tag{A.1}$$

Compared to (1.29), (A.1) does not require the computation of  $\frac{1}{\rho}Z^\ell$  twice per iteration. The scheme (A.1) is known as the scaled ADMM, see e.g., [34, Sect. 3.1.1].

## A.2 Runtimes per MAX-3-SAT instance

We provide the running times of SOS-MS on each MAX-3-SAT instance from the MSE-2016, on 70 and 80 variables. The instances correspond to Figures 6.1 and 6.2 respectively, see Page 186.

The runtimes are reported in Table A.1. Column *m* reports the number of clauses in the corresponding instance. Column *id* reports the instance identifier, via the following scheme: the instance *id*, on  $n \in \{70, 80\}$  variables, with *m* clauses, has as full name `s3v[n]c[m]-id.cnf`. Column **SOS-MS** provides the runtimes in seconds of the SOS-MS algorithm per instance. Column **MSE** from the 70 variable instances also provides the running times of the solver from the MSE that solved the corresponding instance in lowest time. Since the solvers from the MSE ran on slower hardware than SOS-MS, we report the runtimes of SOS-MS on the 70 variable instances after multiplying them by 1.4. Column **CCLS2akms** reports the runtimes of the CCLS2akms algorithm on the 80 variable instances. This algorithm ran on the same hardware as SOS-MS, and thus we report the original runtimes of both the SOS-MS and CCLS2akms algorithms for the 80 variable instances. Note that 80 variable instances were not tested in the MSE-2016, so we are unable to infer what the best MSE runtimes are.

In Table A.1, table entries N/A indicate that the corresponding instance does not exist with 70 or 80 variables. For example, the 70 variable `s3v70c700-6.cnf` does not exist, but the 80 variable `s3v80c700-6.cnf` does.

The instances `s3v70c1500-1.cnf`, `s3v70c1500-4.cnf` and `s3v70c1500-5.cnf` remained unsolved in the MSE-2016. Using SOS-MS, we compute that their optimal values are 1410, 1409 and 1406, respectively.

		Running time per instance (seconds)			
		70 variables		80 variables	
$m$	id	SOS-MS	MSE	SOS-MS	CCLS2akms
700	1	179.74	5.00	441.66	6.98
	2	145.49	7.06	638.47	13.29
	3	298.63	14.14	489.35	4.95
	4	381.82	13.83	144.89	1.52
	5	145.19	4.61	799.49	18.21
	6	N/A		402.94	5.16
800	1	281.03	27.20	923.21	44.68
	2	378.84	81.33	> 1800.00	197.40
	3	125.14	16.76	1174.15	43.04
	4	129.03	15.14	557.18	25.93
	5	118.23	26.16	388.10	35.73
	6	N/A		1123.17	39.26
900	1	280.31	67.52	585.08	81.99
	2	235.61	59.97	> 1800.00	181.21
	3	132.43	53.09	730.48	99.57
	4	219.21	54.46	305.24	67.33
	5	321.44	87.16	423.89	67.35
	6	N/A		298.67	35.94
1000	1	312.57	123.08	1316.13	355.26
	2	193.14	56.66	1307.67	273.86
	3	122.33	78.94	327.52	93.09
	4	100.70	122.22	1559.63	483.51
	5	143.14	47.84	405.84	151.73
	6	N/A		366.26	168.22
1100	1	531.20	331.47	1624.88	937.81
	2	244.50	227.34	> 1800.00	1393.08
	3	122.36	136.06	279.54	344.01
	4	230.81	169.47	1716.55	806.55
	5	176.46	184.75	587.51	264.34
	6	N/A		1660.14	425.80

		Running time per instance (seconds)			
		70 variables		80 variables	
$m$	id	SOS-MS	MSE	SOS-MS	CCLS2akms
1200	1	756.47	752.93	542.94	790.99
	2	161.94	322.68	201.26	333.18
	3	645.56	608.15	592.83	522.42
	4	739.14	1011.74	641.53	486.97
	5	473.66	544.98	1030.56	868.28
	6	N/A		1729.55	1131.22
1300	1	520.31	1139.98	330.12	668.63
	2	541.27	946.71	614.47	1314.20
	3	367.58	708.07	> 1800.00	> 1800.00
	4	153.01	316.19	428.29	992.42
	5	278.93	650.70	> 1800.00	> 1800.00
	6	N/A		1665.65	> 1800.00
1400	1	129.06	705.64	443.53	> 1800.00
	2	854.36	1309.01	872.60	> 1800.00
	3	249.01	990.89	695.00	> 1800.00
	4	184.05	284.52	646.95	1044.26
	5	299.69	1062.63	987.23	> 1800.00
	6	N/A		1433.79	> 1800.00
1500	1	466.85	> 1800.00		
	2	270.72	1182.99		
	3	191.68	893.82		N/A
	4	642.74	> 1800.00		
	5	433.93	> 1800.00		

Table A.1: 70 and 80 variable MAX-3-SAT instances

### A.3 Branching process for the partial MAX-2-SAT problem

During the B&B search for the optimal solution to partial MAX-2-SAT problems, we do not compute SOS-SDP based upper bounds at each node. We describe here the process which decides in which nodes the algorithm computes an upper bound.

Recall that our branching rule, described in Section 6.10.2 on Page 180, selects the variable  $i \in [n]$  for which either  $\text{unitRes}(\phi, i)$  or  $\text{unitRes}(\phi, -i)$  contains many hard unit clauses, and therefore many truth assignments. Each branching step creates two child nodes. We refer to the node which corresponds to the proposition with most truth assignments in the two child nodes as a *bad* node. The forced assignments resulting from the hard unit clauses in that proposition are often sub optimal, which explains the name.

The algorithm for the B&B search operates in two phases, named phase I and phase II. In phase I, we only compute upper bounds for bad nodes. We exit phase I when the algorithm fails to prune a bad node, or when the number of remaining

variables is smaller than some fixed value  $n_{\min}$ . This process is given in pseudocode in Algorithm 5.

After exiting phase I, the algorithm enter phase II, see Algorithm 6. In phase II, before we compute an upper bound in a node, we first attempt to remove variables from the proposition, by pruning bad nodes. This is described in Lines 6 to 12. The main difference with phase I is that, when we fail to prune a bad node in these lines, we do not recompute a stronger upper bound with a larger monomial basis. The extra effort in phase I is justified since pruning a node in phase I equates to removing one unassigned variable from the rest of the search tree.

After Lines 6 to 12 of Algorithm 6, we consider the remaining proposition  $\phi$ , and compute the basis  $SOS_s^1$ . If this basis is too large, we branch immediately. Otherwise, we compute an SDP upper bound using this basis. We set the parameters as  $\theta_{\text{start}} = 0$  in phase I,  $\theta_{\text{start}} = 0.1$  in phase II, and  $(b_{\max}, n_{\min}, s_{\max}) = (15, 70, 1750)$ .

---

**Algorithm 5:** B&B search for the (weighted) partial MAX-2-SAT problem, phase I

---

```

1 Input: Lower bound LB, proposition  $\phi$ , parameters
    $(\theta_{\text{start}}, n_{\min}) \in [0, 0.5) \times \mathbb{N}$ .
2 Set  $\theta = \theta_{\text{start}}$ .
3 while  $n_\phi \geq n_{\min}$  do
4   Determine the truth assignment  $\sigma$  according to the branching rule from
   Section 6.10.2.
5   Compute  $\phi' = \text{unitRes}(\phi, \sigma)$ .
   /* Proposition  $\phi'$  corresponds to a bad node. */
6   Solve  $P_{\phi'}$ , using basis  $SOS_s^\theta$ , to obtain UB.
7   if  $\lfloor \text{UB} \rfloor \leq \text{LB}$  then
8     Update  $\phi := \text{unitRes}(\phi, -\sigma)$ , and reset  $\theta$  by  $\theta := \theta_{\text{start}}$ .
     /* Note that we did not compute an upper bound for the old
        $\phi$ . */
9   else
10    if  $\theta = \theta_{\text{start}}$  then
11      Set  $\theta = 0.5$  if  $\text{UB} - \text{LB} < 10$ , set  $\theta = 1$  otherwise.
      /* Recompute a stronger upper bound with a larger  $\theta$ . */
12    else
      /* Stronger upper bound was unable to prune the node.
        */
13      Compute  $\phi'' = \text{unitRes}(\phi, -\sigma)$ .
14      Add two nodes corresponding to  $\phi'$  and  $\phi''$  to the search tree.
15      break
      /* Move to phase II of the algorithm (see Algorithm 6).
        */

```

---

---

**Algorithm 6:** B&B search for the (weighted) partial MAX-2-SAT problem, phase II

---

```

1 Input: Lower bound LB, parameters
    $(\theta_{\text{start}}, b_{\text{max}}, n_{\text{min}}, s_{\text{max}}) \in [0, 1) \times \mathbb{N} \times \mathbb{N} \times \mathbb{N}$ .
2 while The search tree contains a node which is neither branched nor pruned
   do
3   Consider an unbranched and unpruned node in the search tree, with
     proposition  $\phi$ .
4   Set  $b = 0$ .
5   while  $b < b_{\text{max}} \ \& \ n_{\phi} > n_{\text{min}}$  do
6     Determine the truth assignment  $\sigma$  according to the branching rule
       from Section 6.10.2.
7     Compute  $\phi' = \text{unitRes}(\phi, \sigma)$ .
       /* Proposition  $\phi'$  corresponds to a bad node. */
8     Solve  $P_{\phi'}$ , using basis  $SOS_s^{\theta_{\text{start}}}$ , to obtain UB.
9     if  $\lfloor \text{UB} \rfloor \leq \text{LB}$  then
10      Update  $\phi := \text{unitRes}(\phi, -\sigma)$ , and  $b := b + 1$ .
        /* Note that we did not compute an upper bound for the
          old  $\phi$ . */
11      else
12        break
13    Compute the monomial basis  $SOS_s^1$  for  $\phi$ .
14    if  $|SOS_s^1| > s_{\text{max}}$  then
15      Branch the node corresponding to  $\phi$ , add its two children to the search
        tree and continue with B&B search.
        /* For efficiency reasons, compute upper bounds only when
          the basis is small enough. */
16    else
17      Solve  $P_{\phi}$ , using basis  $SOS_s^1$ , to obtain UB.
18      if  $\lfloor \text{UB} \rfloor \leq \text{LB}$  then
19        Prune the node corresponding to  $\phi$ , and continue with the B&B
          search.
20      else
21        Perform Line 15.

```

---

## A.4 Search tree for the partial MAX-2-SAT problem

We provide the search trees of our SOS-SDP based algorithm for solving various partial MAX-2-SAT instances, as described in Section 6.10.2 on Page 180. These instances are also reported in Tables 6.3 and 6.4.

For the search trees in Figures A.1 to A.3, each node is given a numeric value between zero and one, or the value **B**. Numeric values indicate the largest value of

$\theta$  for which basis  $SOS_s^\theta$  was used to compute an upper bound in that node. The value **B** (**B** for branch) indicates that no upper bound was computed in this node, but instead immediately a variable was chosen to branch.

The figures show the strength of the SDP bounds, implying that many nodes can be pruned immediately. This also demonstrates the effectiveness of the branching rule, which is able to find many nodes that can be pruned immediately.

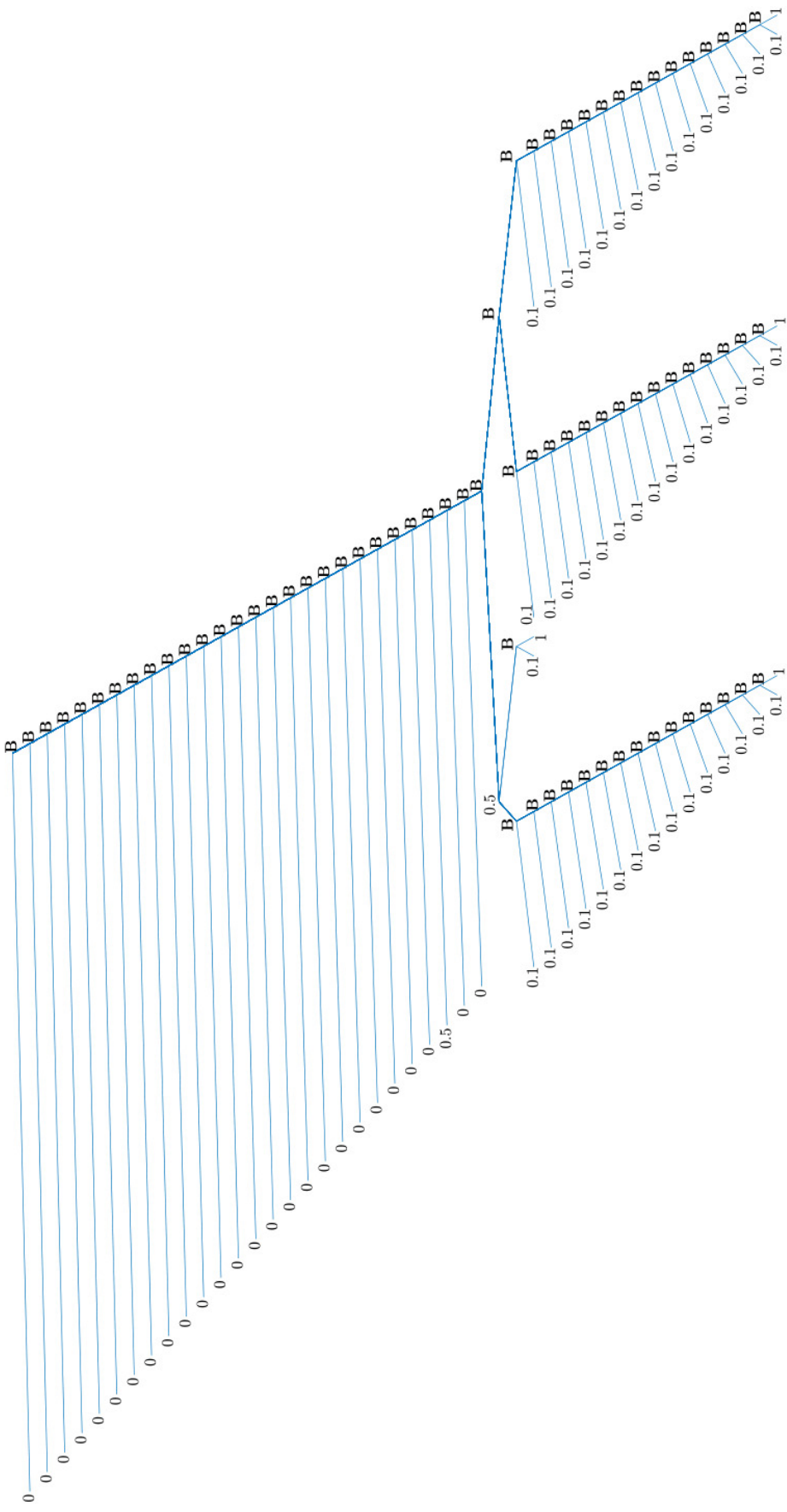


Figure A.1: Search tree for `file_rpms_wcnf_L2_V150_C2500_H150_4.wcnf`

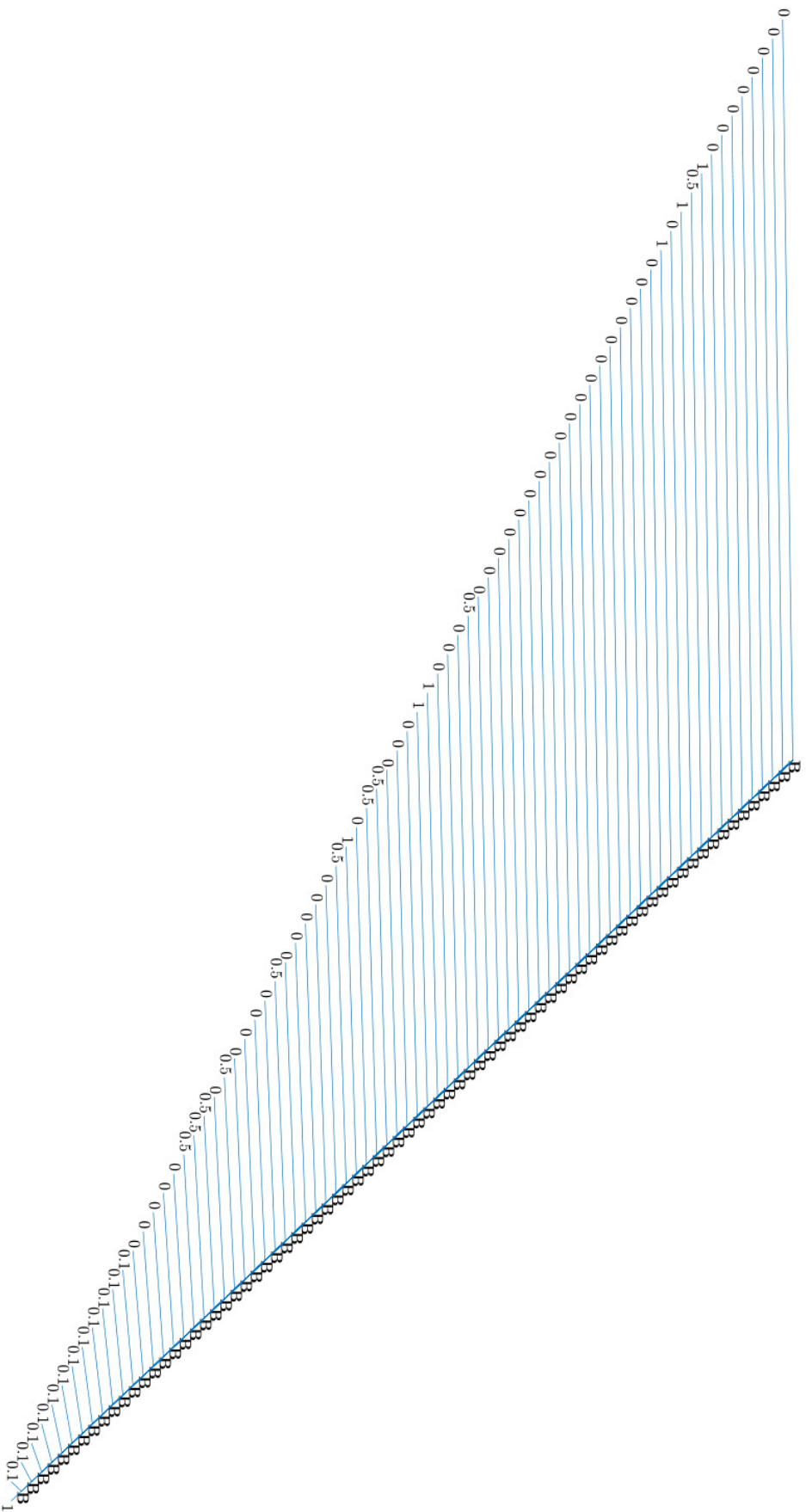


Figure A.2: Search tree for file\_rwpms\_wcnf\_L2\_V150\_C1000\_H150\_1.wcnf



## A.5 Reflection symmetries of $\mathcal{U}_m$ and $\text{CUT}_m^n$

In this section, we prove a claim relating to the extension of reflection symmetries of  $\mathcal{U}_m$  to  $\text{CUT}_m^n$ . This claim was made in Section 3.2.1, Page 61.

Let  $n, m \in \mathbb{N}$ , and let  $\sigma : \mathbb{C} \rightarrow \mathbb{C}$  be a reflection symmetry of  $\mathcal{U}_m$ , see (3.1) on Page 57. That is,  $\sigma$  is a reflection that satisfies  $\sigma(\mathcal{U}_m) = \mathcal{U}_m$ . We extend  $\sigma$  to  $\mathbb{C}^n$  by defining, for  $u \in \mathbb{C}^n$ ,  $\sigma(u)$  via entrywise evaluation of  $\sigma$  to the entries of  $u$ . To extend  $\sigma$  to  $\text{CUT}_m^n$ , note that any  $X \in \text{CUT}_m^n$  can be written as  $X = G^H G$ , for some matrix  $G \in \mathbb{C}^{n \times n}$ , since  $X \succeq 0$ . We then extend  $\sigma$  to  $\text{CUT}_m^n$  by defining

$$\sigma(X) := \sigma(G)^H \sigma(G), \text{ for } \sigma(G) = (\sigma(G_{ij}))_{i,j \in [n]}, \text{ and } X = G^H G. \quad (\text{A.2})$$

**Lemma A.1.** *Let  $n, m \in \mathbb{N}$ , and let  $\sigma : \mathbb{C} \rightarrow \mathbb{C}$  be a reflection symmetry of  $\mathcal{U}_m$ . For any  $X \in \text{CUT}_m^n$ , define  $\sigma(X)$  as in (A.2). We have that  $\sigma(X)$  is well-defined, and  $\sigma(X) = \overline{X}$ .*

*Proof.* Any reflection symmetry of  $\mathcal{U}_m$  can be written as  $\sigma(re^{\phi \mathbf{i}}) = re^{(2\theta - \phi)\mathbf{i}}$ , where  $\theta \in \mathbb{R}$  is the angle between the line of reflection and the axis  $\text{Im}(x) = 0$ . Therefore, for  $x_1, x_2 \in \mathbb{C}$ , with  $x_j = r_j e^{\phi_j \mathbf{i}}$  and  $r_j, \phi_j \in \mathbb{R}$ ,  $j \in [2]$ , we have that

$$\overline{\sigma(x_1)} \sigma(x_2) = r_1 r_2 \overline{e^{(2\theta - \phi_1)\mathbf{i}}} e^{(2\theta - \phi_2)\mathbf{i}} = r_1 r_2 e^{(\phi_1 - \phi_2)\mathbf{i}} = x_1 \overline{x_2}. \quad (\text{A.3})$$

For vectors  $u, v \in \mathbb{C}^n$ , (A.3) immediately shows that

$$\sigma(u)^H \sigma(v) = \sum_{i \in [n]} \overline{\sigma(u_i)} \sigma(v_i) = \sum_{i \in [n]} u_i \overline{v_i} = v^H u. \quad (\text{A.4})$$

To show that  $\sigma(X)$  is well-defined, note that the matrix  $G$  satisfying  $X = G^H G$  is unique up to multiplication on the left by an orthogonal matrix  $Q \in \mathbb{C}^{n \times n}$ , i.e.,  $G \rightarrow QG$ . Therefore, as  $X = G^H G = (QG)^H (QG)$ , it must hold that

$$\sigma(G)^H \sigma(G) = \sigma(QG)^H \sigma(QG). \quad (\text{A.5})$$

To verify (A.5), let vectors  $\mathbf{e}_1, \dots, \mathbf{e}_n \in \{0, 1\}^n$  form the canonical basis of unit vectors of  $\mathbb{R}^n$ . Then, using (A.4), we find that

$$(\sigma(G)^H \sigma(G))_{ij} = \sigma(G \mathbf{e}_i)^H \sigma(G \mathbf{e}_j) = \mathbf{e}_j^\top G^H G \mathbf{e}_i = X_{ji}. \quad (\text{A.6})$$

For  $Q$  an orthogonal matrix, i.e.,  $Q^H Q = \mathbf{I}$ , we again use (A.4) to derive

$$(\sigma(QG)^H \sigma(QG))_{ij} = \sigma(QG \mathbf{e}_i)^H \sigma(QG \mathbf{e}_j) = \mathbf{e}_j^\top G^H Q^H Q G \mathbf{e}_i = X_{ji}. \quad (\text{A.7})$$

The combination of (A.6) and (A.7) proves (A.5), so that  $\sigma(X)$  is well-defined.

Equation (A.6) also shows that  $\sigma(X)_{ij} = X_{ji}$ . Since  $X$  is Hermitian,  $X_{ji} = \overline{X_{ij}}$ , so that  $\sigma(X) = \overline{X}$ , which completes the proof.  $\square$

## A.6 Facet enumeration of $\mathcal{V}(\text{CUT}_3^3)$

Table A.2 provides a complete enumeration of the 27 facet defining inequalities of  $\mathcal{V}(\text{CUT}_3^3)$ , see (3.28). These inequalities are given implicitly in Theorem 3.17 on Page 70, and one inequality is given explicitly in Proposition 3.12 on Page 68. The facet defining inequalities in Table A.2 were verified using the SageMath software [295]. The corresponding SageMath code is provided in Listing A.1.

In Table A.2, inequalities 1 up to and including 18 are the triangle facets, using the terminology of Section 3.3. Inequalities 19 up to and including 27 are the inequalities that ensure  $x_i \in \text{Conv}(\mathcal{B}_3)$  for all  $i \in [3]$ , see (3.6).

```
# Create the field extension to symbolically evaluate sqrt(3)
x = polygen(ZZ, 'x')
K.<sqrt3> = NumberField(x^2 - 3, embedding=AA(3)**(1/2))
r = sqrt3/2

# Create the polyhedron with points of the cut polytope.
# Points are given as (Re(x), Im(x)).
P = Polyhedron([(1,1,1,0,0,0), (1,-0.5,-0.5,0,-r,-r),
(1,-0.5,-0.5,0,r,r), (-0.5,1,-0.5,-r,0,r),
(-0.5,-0.5,1,-r,-r,0), (-0.5,-0.5,-0.5,-r,r,-r),
(-0.5,1,-0.5,r,0,-r), (-0.5,-0.5,-0.5,r,-r,r),
(-0.5,-0.5,1,r,r,0)])

# Compute the facet defining inequalities (i.e.,
# half-space representation) of the cut polytope P
P.Hrepresentation()
```

Listing A.1: SageMath code for computing the facets of  $\mathcal{V}(\text{CUT}_3^3)$ .

## A.7 Computational details of proof of Lemma 4.17

We provide computational details on the proof of Lemma 4.17 on Page 107. Specifically, the verification of  $c(G, s) \leq \lfloor s/2 \rfloor$  for all  $G \in \mathbb{G}_s$  for  $s \in \{5, 7, 9, 11, 13\}$ . We performed the following steps on a laptop (16 GB RAM and Intel i7-1165G7 CPU), which required approximately 16 hours to run. Our code is available at

[https://github.com/LMSinjorgo/QMC\\_proofVerification](https://github.com/LMSinjorgo/QMC_proofVerification).

We first use the software package `nauty` [222] to generate the graphs in  $\mathbb{G}_s$  that satisfy the properties 1 to 3 of Lem. 4.16. Then, for the case  $s \in \{5, 7, 9\}$ , we verify that  $c(G, 2) \leq \lfloor s/2 \rfloor$  for these graphs in  $\mathbb{G}_s$  using SDP. Computing  $c(G, 2)$  can be done using the Pauli-based  $\text{SDP}^k$ , but, in fact, solving a relaxation of  $\text{SDP}^k$  based on the SWAP operators already suffices to establish the desired upper bound. (More precisely, we compute the first level of the QMC SDP relaxation based on the SWAP operators, see [293, 311] and in particular the discussion in [293, Sect. 5.1.2].)

For  $s = 11$ , we consider the 26360 (see Table 4.3 on Page 107) remaining graphs  $G_j = ([11], E_j)$ ,  $j \in [26360]$  in the sequence as returned by `nauty`. We construct the

	$\eta_1$	$\eta_2$	$\eta_3$
1.	$\mathbf{i}$	$e^{\pi\mathbf{i}/6}$	$\mathbf{i}$
2.	$e^{5\pi\mathbf{i}/6}$	$e^{-\pi\mathbf{i}/6}$	$e^{\pi\mathbf{i}/6}$
4.	$e^{\pi\mathbf{i}/6}$	$e^{-\pi\mathbf{i}/6}$	$e^{5\pi\mathbf{i}/6}$
5.	$\mathbf{i}$	$-\mathbf{i}$	$e^{-\pi\mathbf{i}/6}$
5.	$e^{\pi\mathbf{i}/6}$	$e^{-5\pi\mathbf{i}/6}$	$e^{\pi\mathbf{i}/6}$
6.	$e^{-\pi\mathbf{i}/6}$	$-\mathbf{i}$	$\mathbf{i}$
7.	$e^{-5\pi\mathbf{i}/6}$	$e^{\pi\mathbf{i}/6}$	$e^{-\pi\mathbf{i}/6}$
8.	$-\mathbf{i}$	$\mathbf{i}$	$e^{\pi\mathbf{i}/6}$
9.	$e^{\pi\mathbf{i}/6}$	$\mathbf{i}$	$-\mathbf{i}$
10.	$e^{-\pi\mathbf{i}/6}$	$e^{\pi\mathbf{i}/6}$	$e^{-5\pi\mathbf{i}/6}$
11.	$e^{-\pi\mathbf{i}/6}$	$e^{5\pi\mathbf{i}/6}$	$e^{-\pi\mathbf{i}/6}$
12.	$-\mathbf{i}$	$e^{-\pi\mathbf{i}/6}$	$-\mathbf{i}$
13.	$e^{5\pi\mathbf{i}/6}$	$\mathbf{i}$	$e^{5\pi\mathbf{i}/6}$
14.	$e^{-5\pi\mathbf{i}/6}$	$e^{5\pi\mathbf{i}/6}$	$\mathbf{i}$
15.	$\mathbf{i}$	$e^{5\pi\mathbf{i}/6}$	$e^{-5\pi\mathbf{i}/6}$
16.	$e^{5\pi\mathbf{i}/6}$	$e^{-5\pi\mathbf{i}/6}$	$-\mathbf{i}$
17.	$e^{-5\pi\mathbf{i}/6}$	$-\mathbf{i}$	$e^{-5\pi\mathbf{i}/6}$
18.	$-\mathbf{i}$	$e^{-5\pi\mathbf{i}/6}$	$e^{5\pi\mathbf{i}/6}$
19.	$\sqrt{3}e^{\pi\mathbf{i}/3}$	0	0
20.	$\sqrt{3}e^{-\pi\mathbf{i}/3}$	0	0
21.	$-\sqrt{3}$	0	0
22.	0	$\sqrt{3}e^{-\pi\mathbf{i}/3}$	0
23.	0	$\sqrt{3}e^{\pi\mathbf{i}/3}$	0
24.	0	$-\sqrt{3}$	0
25.	0	0	$\sqrt{3}e^{\pi\mathbf{i}/3}$
26.	0	0	$\sqrt{3}e^{-\pi\mathbf{i}/3}$
27.	0	0	$-\sqrt{3}$

Table A.2: Coefficients  $\eta$  of the facet defining inequalities  $\operatorname{Re}\left(\sum_{i=1}^3 \eta_i x_i\right) \leq \frac{\sqrt{3}}{2}$  for  $\mathcal{V}(\operatorname{CUT}_3^3)$ .

corresponding Hamiltonians recursively as

$$H_{G_{j+1}} = H_{G_j} + \sum_{e \in E_{j+1} \setminus E_j} H_e - \sum_{e \in E_j \setminus E_{j+1}} H_e. \quad (\text{A.8})$$

Constructing  $H_{G_{j+1}}$  using (A.8) is efficient since `nauty` returns a sequence of graphs where  $E_j \approx E_{j+1}$ . Additionally, (A.8) shows that

$$\lambda_{\max}(H_{G_{j+1}}) \leq \lambda_{\max}(H_{G_j}) + \lambda_{\max}\left(\sum_{e \in E_{j+1} \setminus E_j} H_e\right). \quad (\text{A.9})$$

Here, we have used that  $H_e = (1/4)H_e^2 \succeq 0$ . Generally,  $\lambda_{\max}\left(\sum_{e \in E_{j+1} \setminus E_j} H_e\right) \leq 4|E_{j+1} \setminus E_j|$ , and tighter bounds are possible if, for example, the edges  $E_{j+1} \setminus E_j$  form a star graph. If (A.9) already proves that  $c(G_{j+1}, s) = \lambda_{\max}(H_{G_{j+1}})/2 - |E_{j+1}| \leq \lfloor s/2 \rfloor$ , we do not carry out the computation of  $\lambda_{\max}(H_{G_{j+1}})$ , nor the construction of  $H_{G_{j+1}}$ . If the bound (A.9) does not prove  $c(G_{j+1}, s) \leq \lfloor s/2 \rfloor$ , we compute  $\lambda_{\max}(H_{G_{j+1}})$  with MATLAB's `eigs` function.

The case  $s = 13$  proceeds similarly as the case  $s = 11$ , except we first discard some of the 9035088 graphs in  $\mathbb{G}_{13}$  that satisfy properties 1 to 3 of Lem. 4.16, by arguing as follows: of these 9035088 graphs, 959842 of them do not satisfy property 4 with  $|S| = 2$ , and so we may discard those. We verify that these 959842 graphs do not satisfy property 4 in approximately 5 seconds. Of the now remaining 8075246 graphs, 1622184 of them satisfy  $\tau(G) \leq 6$ . We verify that these 1622184 graphs satisfy  $\tau(G) \leq 6$  in approximately 20 seconds. By Item 2 of Lem. 4.15, these graphs satisfy  $c(G, 13) \leq \tau(G) \leq 6$ , so we may discard them as well. For the remaining 6453062 graphs  $G$ , we compute upper bounds on  $\lambda_{\max}(H_G)$  in the manner described for the case  $s = 11$ .

# B Technical lemmas and proofs

## B.1 Proofs from Chapter 2

The following known result (cf. e.g. [297]) is used in the proofs of Theorem 2.20 and Lemma 2.23, on Pages 35 and 38 respectively. We provide a proof for the sake of completeness.

**Lemma B.1.** *Let  $g_1, \dots, g_n$ ,  $n \in \mathbb{N}$ , be convex functions from  $\mathbb{R}^d$  to  $\mathbb{R}$ , for some  $d \in \mathbb{N}$ . For some  $k \in [n]$ , define  $g(x) := \mathbf{S}_k(\{g_1(x), \dots, g_n(x)\})$  as the function that returns the sum of the  $k$  largest values in  $\{g_1(x), \dots, g_n(x)\}$ ,  $x \in \mathbb{R}^d$ . Function  $g$  is convex.*

*Proof.* Let  $x, x' \in \mathbb{R}^d$  and  $w \in [0, 1]$ . We have

$$\begin{aligned} wg(x) + (1-w)g(x') &= w\mathbf{S}_k(\{g_1(x), \dots, g_n(x)\}) + (1-w)\mathbf{S}_k(\{g_1(x'), \dots, g_n(x')\}) \\ &= \mathbf{S}_k(\{wg_1(x), \dots, wg_n(x)\}) + \mathbf{S}_k(\{(1-w)g_1(x'), \dots, (1-w)g_n(x')\}) \\ &\geq \mathbf{S}_k(\{wg_1(x) + (1-w)g_1(x'), \dots, wg_n(x) + (1-w)g_n(x')\}) \\ &\geq \mathbf{S}_k(\{g_1(wx + (1-w)x'), \dots, g_n(wx + (1-w)x')\}) \\ &= g(wx + (1-w)x'). \end{aligned} \quad \square$$

## B.2 Proofs from Chapter 3

We require the following result to prove Lemma 3.14 on Page 69.

**Lemma B.2.** *Let  $r := (2 \cos \frac{\pi}{9})^{-1} \approx 0.53$  and  $y = e^{2\pi i/9}$ . We have that*

$$Y := \begin{bmatrix} 1 & ry^2 & ry \\ r\bar{y}^2 & 1 & ry^2 \\ r\bar{y} & r\bar{y}^2 & 1 \end{bmatrix} \in \mathcal{E}_3^3. \quad (\text{B.1})$$

*Proof.* It is easily verified that the values  $ry^2$ ,  $ry$ , and their complex conjugates, are contained in  $\text{Conv}(\mathcal{U}_3)$ . It remains to show that  $Y \succeq 0$ , which we do by showing that the principal submatrices of  $Y$  have nonnegative determinants.

All  $1 \times 1$  principal submatrices of  $Y$  have determinant 1. All  $2 \times 2$  principal submatrices of  $Y$  have determinant  $1 - r^2 \approx 0.72$ . To compute these determinants,

we have used that  $|y| = 1$ . Lastly,

$$\begin{aligned} \det(Y) &= r^3 (y^3 + \bar{y}^3) - 3r^2 + 1 = 2r^3 \cos\left(\frac{6\pi}{9}\right) - 3r^2 + 1 \\ &= \frac{\cos\left(\frac{6\pi}{9}\right) - 3\cos\left(\frac{\pi}{9}\right) + 4\cos^3\left(\frac{\pi}{9}\right)}{4\cos^3\left(\frac{\pi}{9}\right)}. \end{aligned} \tag{B.2}$$

Substituting the triple angle identity  $4\cos^3(x) = 3\cos(x) + \cos(3x)$  in (B.2) shows that the numerator of (B.2) equals  $\cos\left(\frac{6\pi}{9}\right) + \cos\left(\frac{3\pi}{9}\right) = 0$ , so that  $\det(Y) = 0$ .  $\square$

*Proof of Lemma 3.14, Page 69.* Since inequality (3.29) is facet defining, the inequality is tight. It therefore follows that  $\max_{X \in \text{CUT}_3^3} \langle Q, X \rangle = \frac{\sqrt{3}}{2}$ . We now claim that

$$Q^* := \max_{X \in \mathcal{E}_3^3} \langle Q, X \rangle = \frac{3\cos\left(\frac{\pi}{18}\right)}{2\cos\left(\frac{\pi}{9}\right)} \approx 1.57, \tag{B.3}$$

which would prove that  $\text{str}(Q, 3) = Q^* / \max_{X \in \text{CUT}_3^3} \langle Q, X \rangle = \frac{\sqrt{3}\cos\left(\frac{\pi}{18}\right)}{\cos\left(\frac{\pi}{9}\right)}$ .

We now prove (B.3). For any  $Y \in \mathcal{E}_3^3$ , the value  $\langle Q, Y \rangle$  provides a lower bound on  $Q^*$ . In particular, for  $Y$  as in (B.1), we have that

$$\max_{X \in \mathcal{E}_3^3} \langle Q, X \rangle \geq \langle Q, Y \rangle = Q^*. \tag{B.4}$$

We now derive the matching upper bound. For any  $X \in \mathcal{E}_3^3$ , the inner product  $\langle Q, X \rangle$  can be rewritten as follows, for  $r := (2\cos\frac{\pi}{9})^{-1} \approx 0.53$  and  $q := (4\sin(2\pi/9))^{-1} \approx 0.39$ :

$$\begin{aligned} \langle Q, X \rangle &= Q^* - \left(\sqrt{3} - \frac{2q}{r}\right) \sum_{1 \leq i < j \leq 3} \text{Re}\left(\frac{1}{2} - e^{\pi i/3} X_{ij}\right) \\ &\quad - q \left\langle \begin{bmatrix} 1 & e^{-4\pi i/9} & e^{-8\pi i/9} \\ e^{4\pi i/9} & 1 & e^{-4\pi i/9} \\ e^{8\pi i/9} & e^{4\pi i/9} & 1 \end{bmatrix}, X \right\rangle \leq Q^* \quad \forall X \in \mathcal{E}_3^3, \end{aligned} \tag{B.5}$$

The inequality in (B.5) is due to the following facts:  $q > 0$ , the matrix

$$\begin{bmatrix} 1 & e^{-4\pi i/9} & e^{-8\pi i/9} \\ e^{4\pi i/9} & 1 & e^{-4\pi i/9} \\ e^{8\pi i/9} & e^{4\pi i/9} & 1 \end{bmatrix} = \begin{bmatrix} 1 \\ e^{4\pi i/9} \\ e^{8\pi i/9} \end{bmatrix} \begin{bmatrix} 1 \\ e^{4\pi i/9} \\ e^{8\pi i/9} \end{bmatrix}^H \succeq 0,$$

the term  $\text{Re}\left(\frac{1}{2} - e^{\pi i/3} X_{ij}\right) \geq 0$  (since  $X_{ij} \in \text{Conv } \mathcal{U}_3$ ) and  $\sqrt{3} - \frac{2q}{r} \geq 0$ . Claim (B.3) follows from combining (B.4) and (B.5), which completes the proof.  $\square$

The following definition is used in the proof of Theorem 3.17 on Page 70.

**Definition B.3.** A convex subset  $F$  of a convex set  $C$  is said to be a face of  $C$  if it satisfies the following: if  $x, y \in C$  and  $t \in (0, 1)$  are such that  $tx + (1 - t)y \in F$ , then  $x, y \in F$ .

*Proof of Lemma 3.33, Page 83.* Proof by induction: the statement is trivially true for  $r = 1$ , since rank one matrices in  $\mathcal{E}_\infty^n$  are extreme points of  $\mathcal{E}_\infty^n$  themselves. Assume the result holds up to  $r - 1$ , for some  $r \geq 2$ . It remains to prove the result for  $A \in \mathcal{E}_\infty^n$  with  $\text{rk}(A) = r$ . If  $A$  is an extreme point of  $\mathcal{E}_\infty^n$ , we are done, since then  $A \in \text{Conv}\{A\}$ , and  $\text{rk}(A) \leq \text{rk}(A)$ .

Suppose that  $A$  is not an extreme point of  $\mathcal{E}_\infty^n$ . Let  $A_1$  be an extreme point of  $\mathcal{E}_\infty^n$  for which the value

$$t^* := \max \{t : (1 - t)A_1 + tA \in \mathcal{E}_\infty^n\}$$

satisfies  $t^* > 1$ . Such a matrix  $A_1$  exists because  $A$  is not an extreme point of  $\mathcal{E}_\infty^n$ . Set  $A_2 := (1 - t^*)A_1 + t^*A$  and note that  $A_2 \neq A$ . Matrix  $A$  lies on the line segment with endpoints  $A_1$  and  $A_2$ , i.e.,  $A \in \text{Conv}\{A_1, A_2\}$ .

Let us now write  $A = G^H G$  for some  $G \in \mathbb{C}^{r \times n}$  with  $\text{rk}(G) = r$ . Consider the perturbation  $B = G^H R G$ ,  $R \in \mathcal{H}^r$ , of  $A$  that satisfies  $A + t_i B = A_i$ , for all  $i \in \{1, 2\}$ , and for some real numbers  $t_1 > 0$  and  $t_2 < 0$ . Then

$$A_i = G^H (\mathbf{I}_r + t_i R) G, \quad i \in \{1, 2\}. \quad (\text{B.6})$$

Matrix  $A_1$  is an extreme point of  $\mathcal{E}_\infty^n$ , and therefore lies on the boundary of  $\mathcal{E}_\infty^n$ . By optimality of  $t^*$ , matrix  $A_2$  also lies on the boundary  $\mathcal{E}_\infty^n$ . Hence, the matrices  $A_i$ ,  $i \in \{1, 2\}$ , are both PSD, but not positive definite. Therefore, as  $G$  is full rank and the  $A_i$  satisfy (B.6), it follows that the matrices  $\mathbf{I}_r + t_i R$ ,  $i \in \{1, 2\}$  are also PSD, but not positive definite. In particular, this implies that  $\text{rk}(\mathbf{I}_r + t_i R) \leq r - 1$ , so that

$$\text{rk}(A_i) = \text{rk}(G^H (\mathbf{I}_r + t_i R) G) \leq \text{rk}(\mathbf{I}_r + t_i R) \leq r - 1 < \text{rk}(A).$$

We have shown that  $A \in \text{Conv}\{A_1, A_2\}$ , where  $A_1$  is an extreme point of  $\mathcal{E}_\infty^n$ , and both matrices  $A_1, A_2$ , have rank strictly smaller than  $\text{rk}(A) = r$ . By the induction hypothesis, matrix  $A_2$  is the convex combination of at most  $r - 1$  extreme points of  $\mathcal{E}_\infty^n$ . That is, there exist extreme points  $\tilde{A}_1, \dots, \tilde{A}_{r-1}$  of rank at most  $r - 1$ , such that  $A_2 \in \text{Conv}\{\tilde{A}_1, \dots, \tilde{A}_{r-1}\}$ . This completes the proof, since

$$A \in \text{Conv}\{A_1, A_2\} \subseteq \text{Conv}\{A_1, \tilde{A}_1, \dots, \tilde{A}_{r-1}\},$$

and the ranks of the matrices  $A_1, \tilde{A}_1, \dots, \tilde{A}_{r-1}$  are strictly smaller than  $\text{rk}(A)$ .  $\square$

*Proof of Lemma 3.35, Page 83.* The case  $r = 2$  follows from Theorem 3.34. We assume that  $r \in \{3, 4\}$  and write

$$A = B^H B, \text{ for some } B = [b_1 \quad b_2 \quad b_3 \quad b_4], \quad b_i \in \mathbb{C}^r, \quad \|b_i\| = 1 \quad \forall i \in [4]. \quad (\text{B.7})$$

Note that, given  $A$ , the matrix  $B$  is unique up to unitary multiplication, i.e.,  $B \rightarrow QB$ , for  $Q$  a unitary matrix in  $\mathbb{C}^{r \times r}$ . We claim that there exists a unitary  $Q \in \mathbb{C}^{r \times r}$  such that

$$QB = \begin{bmatrix} \sqrt{w} q^H \\ \sqrt{1 - w} G \end{bmatrix}, \text{ for } w \in (0, 1), \quad q \in \mathcal{U}_\infty^4, \quad G \in \mathbb{C}^{(r-1) \times 4}, \quad Q^H Q = \mathbf{I}_4. \quad (\text{B.8})$$

If so, (B.8) implies that

$$A = (QB)^H(QB) = wqq^H + (1-w)G^HG \implies A \in \text{Conv}\{qq^H, G^HG\}, \quad (\text{B.9})$$

where  $qq^H$  is a rank 1 extreme point of  $\text{CUT}_\infty^4$ , and  $G^HG \in \mathcal{E}_\infty^4$  with  $\text{rk}(G^HG) = r - 1$ . Note that  $G^HG \in \mathcal{E}_\infty^4$  follows from the fact that  $G^HG \succeq 0$  and  $\text{diag}(G^HG) = \frac{1}{1-w}\text{diag}(A - wqq^H) = \mathbf{1}_4$ .

We now prove the existence of a unitary  $Q$  satisfying (B.8). Consider the following system of equations in a Hermitian matrix variable  $X$ :

$$\langle b_1b_1^H - b_2b_2^H, X \rangle = \langle b_1b_1^H - b_3b_3^H, X \rangle = \langle b_1b_1^H - b_4b_4^H, X \rangle = 0. \quad (\text{B.10})$$

Since  $\|b_i\| = 1$ , see (B.7),  $X = \mathbf{I}_r$  is a solution to (B.10). Then, [6, Thm. 2.2] states that there exists a nonzero vector  $x \in \mathbb{C}^r$  such that  $X = xx^H$  solves (B.10). As the set of solutions to (B.10) is closed under scalar multiplication, we may assume that  $\|x\| = 1$ .

Let  $Q$  be a unitary matrix with  $x^H$  as its first row. Then the first row of  $QB$  consists of the values  $x^Hb_i$ ,  $i \in [4]$ . Since  $X = xx^H$  satisfies (B.10), it follows that  $|x^Hb_i| = |x^Hb_j|$  for all  $i, j \in [4]$ . Thus,  $Q$  satisfies (B.8) and our claim is proven.

It follows, considering (B.9), that  $A \in \text{Conv}\{qq^H, G^HG\}$ . If  $r = \text{rk}(A) = 3$  and  $G^HG$  is an extreme point of  $\mathcal{E}_\infty^4$  we are done. If  $r = 3$  and  $G^HG$  is not an extreme point of  $\mathcal{E}_\infty^4$ , then  $G^HG$  itself is the convex combination of two rank 1 extreme points of  $\mathcal{E}_\infty^4$  by Theorem 3.34. This proves the case  $r = 3$ , which we inductively use to proof the case  $r = 4$ .

If  $r = 4$ , then  $\text{rk}(G^HG) = 3$ , and so  $G^HG$  is not an extreme point of  $\mathcal{E}_\infty^4$  by Lemma 3.21. Then, using Lemma 3.35 for  $r = 3$ , there exist extreme points  $A_1, \dots, A_k$ ,  $k \leq 3$ , satisfying  $G^HG \in \text{Conv}\{A_1, \dots, A_k\}$  and  $\sum_{i \in [k]} \text{rk}(A_i) = 3$ . Hence,  $A \in \text{Conv}\{qq^H, A_1, \dots, A_k\}$ , which proves the result for  $r = 4$ .  $\square$

We require the following result to prove Lemma 3.38 on Page 84.

**Lemma B.4.** *For the matrix  $H$  as in (3.68), we have that  $-\mathbf{I}_4 \preceq \frac{1}{\sqrt{3}}H \preceq \mathbf{I}_4$ .*

*Proof.* Since  $H^2 = 3\mathbf{I}_4$ , the eigenvalues of  $H$  are contained in  $\{\pm\sqrt{3}\}$ .  $\square$

*Proof of Lemma 3.38, Page 84.* Let  $m \geq 3$  be an integer or  $m = \infty$ . To prove Item 1 of Lemma 3.38, we derive first

$$\max_{X \in \mathcal{E}_m^4} \langle H, X \rangle = \max_{X \in \mathcal{E}_m^4} 4\sqrt{3} - \langle \sqrt{3}\mathbf{I}_4 - H, X \rangle \leq 4\sqrt{3}. \quad (\text{B.11})$$

The equality in (B.11) follows from the fact that  $\langle \sqrt{3}\mathbf{I}_4, X \rangle = 4\sqrt{3}$ . The inequality in (B.11) follows from the fact that  $\sqrt{3}\mathbf{I}_4 - H \succeq 0$  by Lemma B.4, and  $X \succeq 0$  since  $X \in \mathcal{E}_m^4$ . To prove that  $\max_{X \in \mathcal{E}_m^4} \langle H, X \rangle \geq 4\sqrt{3}$ , we consider the matrix

$$Y := rH + \mathbf{I}_4, \text{ for } r := \frac{1}{\sqrt{3}} \approx 0.58.$$

We claim that  $Y \in \mathcal{E}_m^4$ , which implies that  $\max_{X \in \mathcal{E}_m^4} \langle H, X \rangle \geq \langle H, Y \rangle = r \langle H, H \rangle = 4\sqrt{3}$ . Combined with (B.11), this would prove Item 1.

We have that  $Y \succeq 0$  by Lemma B.4. Thus, to prove that  $Y \in \mathcal{E}_m^4$ , it remains to show that the entries of  $Y$  are contained in  $\text{Conv}(\mathcal{U}_m)$ . Note that the entries of  $Y$  are contained in the set  $\{1, r, \pm r\mathbf{i}\}$ . Clearly, 1 and  $r$  are contained in  $\text{Conv}(\mathcal{U}_m)$ . For  $m = 3$ ,  $r\mathbf{i} \in \text{Conv}\{1, e^{2\pi\mathbf{i}/3}\} \subseteq \text{Conv}(\mathcal{U}_3)$  and  $-r\mathbf{i} \in \text{Conv}\{1, e^{4\pi\mathbf{i}/3}\} \subseteq \text{Conv}(\mathcal{U}_3)$ . For  $m = 4$ , note that the polytope  $\text{Conv}(\mathcal{U}_4)$  contains an inscribed circle of radius

$$\min_{x \in \partial \text{Conv}(\mathcal{U}_4)} |x| = \left| \frac{1 + \mathbf{i}}{2} \right| = \frac{1}{\sqrt{2}}. \quad (\text{B.12})$$

Since

$$|\pm r\mathbf{i}| = r = \frac{1}{\sqrt{3}} < \frac{1}{\sqrt{2}}, \quad (\text{B.13})$$

it follows that  $\pm r\mathbf{i} \in \text{Conv}(\mathcal{U}_4)$ . Now for the case  $m > 4$ : let  $R_m$  denote the radius of the inscribed circle of  $\text{Conv}(\mathcal{U}_m)$  (by (B.12),  $R_4 = 1/\sqrt{2}$ ). Note that  $R_m$  is increasing in  $m$ . Therefore, following (B.13), we have

$$|\pm r\mathbf{i}| < R_4 \leq R_m \Rightarrow \{\pm r\mathbf{i}\} \subseteq \text{Conv } \mathcal{U}_m \quad \forall m \geq 4.$$

Thus, we have shown that all elements of  $Y$  are contained in  $\text{Conv}(\mathcal{U}_m)$  for all valid  $m$ . Therefore,  $Y \in \mathcal{E}_m^4$ .

To prove Item 2 of Lemma 3.38, we use that  $\text{CUT}_m^4 \subseteq \text{CUT}_\infty^4 \subseteq \mathbf{L}(\mathcal{B}_1)$ , see (3.48), which implies that

$$\max_{X \in \text{CUT}_m^4} \langle H, X \rangle \leq \max_{X \in \text{CUT}_\infty^4} \langle H, X \rangle \leq \max_{X \in \mathbf{L}(\mathcal{B}_1)} \langle H, X \rangle. \quad (\text{B.14})$$

It follows from (B.14) that we may prove Item 2 of Lemma 3.38 by proving the following two inequalities:  $\max_{X \in \text{CUT}_m^4} \langle H, X \rangle \geq 6$  and  $\max_{X \in \mathbf{L}(\mathcal{B}_1)} \langle H, X \rangle \leq 6$ .

For the first inequality, we use that  $\mathbf{J}_4 \in \text{CUT}_m^4$ , which implies that

$$\max_{X \in \text{CUT}_m^4} \langle H, X \rangle \geq \langle H, \mathbf{J}_4 \rangle = 6.$$

For the second inequality, let  $X \in \mathbf{L}(\mathcal{B}_1)$ , and let  $Z \in \mathcal{F}(\mathcal{B}_1)$  be a matrix satisfying  $Z_{1:4,1:4} = X$ , see (3.44). We have that  $\langle H, X \rangle = 6 - \langle Q, Z \rangle$ , where

$$Q = \frac{1}{2} \begin{bmatrix} 4 & 0 & -2\mathbf{i} & -2 & 2\mathbf{i} & -2 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 2\mathbf{i} & 0 & 2 & -1 - \mathbf{i} & -1 - \mathbf{i} & 0 \\ -2 & 0 & -1 + \mathbf{i} & 2 & 0 & 1 - \mathbf{i} \\ -2\mathbf{i} & 0 & -1 + \mathbf{i} & 0 & 2 & -1 + \mathbf{i} \\ -2 & 0 & 0 & 1 + \mathbf{i} & -1 - \mathbf{i} & 2 \end{bmatrix}.$$

We claim that  $Q \succeq 0$ . Then, since also  $Z \succeq 0$ , we have  $\langle H, X \rangle = 6 - \langle Q, Z \rangle \leq 6$  for any  $X \in \mathbf{L}(\mathcal{B}_1)$ . To show that  $Q \succeq 0$ , we compute the Schur complement of  $Q$  with

respect to  $Q_{11} = 2$ . The resulting matrix is given by

$$\frac{1}{2} \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & -\mathbf{i} & \mathbf{i} \\ 0 & -1 & 1 & \mathbf{i} & -\mathbf{i} \\ 0 & \mathbf{i} & -\mathbf{i} & 1 & -1 \\ 0 & -\mathbf{i} & \mathbf{i} & -1 & 1 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 0 \\ \mathbf{i} \\ -\mathbf{i} \\ -1 \\ 1 \end{bmatrix} \begin{bmatrix} 0 \\ \mathbf{i} \\ -\mathbf{i} \\ -1 \\ 1 \end{bmatrix}^H \succeq 0.$$

□

*Proof of Lemma 3.41, Page 85.* We fix some  $n \geq 2$ . For notational convenience, we omit the superscript  $n$  in sets  $\widetilde{\mathcal{A}}^n$  and  $\mathcal{A}^n$ . It suffices to show that  $\widetilde{\mathcal{A}} \models \mathcal{A}$ , see Definition 3.23. Define the sets

$$\mathcal{D}_0 := \widetilde{\mathcal{A}}, \text{ and } \mathcal{D}_i := \mathcal{D}_0 \cup \{2\mathbf{e}_1, 2\mathbf{e}_2, \dots, 2\mathbf{e}_i\} \text{ for } i \in [n-1],$$

where the vectors  $\mathbf{e}_i \in \{0, 1\}^{n-1}$ ,  $i \in [n-1]$ , are the columns of  $\mathbf{I}_{n-1}$ . Note that  $\mathcal{D}_{n-1} = \mathcal{A}$ . It suffices to show that  $\mathcal{D}_i \models \mathcal{D}_{i+1}$  for all  $i \in \{0, 1, \dots, n-2\}$ . Indeed, by the transitivity of  $\models$ , see Definition 3.23, it then follows that  $\widetilde{\mathcal{A}} = \mathcal{D}_0 \models \mathcal{D}_{n-1} = \mathcal{A}$ .

To prove that  $\mathcal{D}_i \models \mathcal{D}_{i+1}$ , we use again PSD completion theory [125]. Specifically, we show that any  $X' \in \mathcal{F}(\mathcal{D}_i)$  can be extended to an  $X \in \mathcal{F}(\mathcal{D}_{i+1})$  that satisfies  $X_{1:|\mathcal{D}_i|, 1:|\mathcal{D}_i|} = X'$ . To do so, let  $X' \in \mathcal{F}(\mathcal{D}_i)$  and let  $X$  be the associated partially specified matrix, similar as to (3.49). We show that the PSD completion problem corresponding to  $X$  is always feasible, i.e.,  $X$  can always be completed to a PSD matrix in  $\mathcal{F}(\mathcal{D}_{i+1})$  that satisfies  $X_{1:|\mathcal{D}_i|, 1:|\mathcal{D}_i|} = X'$ .

We denote the graph associated to this PSD completion problem by  $\mathcal{G}$ , with as vertices the elements of  $\mathcal{D}_{i+1}$ , and edge set  $E = \{\{\gamma, \beta\} : X_{\gamma, \beta} \neq \star\}$ . Note that  $E$  can be rewritten as

$$E = \{\{\gamma, \beta\} : \gamma, \beta \in \mathcal{D}_{i+1}, \gamma - \beta \in \mathcal{D}_i - \mathcal{D}_i\},$$

where  $\mathcal{D}_i - \mathcal{D}_i := \{\gamma - \beta : \gamma, \beta \in \mathcal{D}_i\}$ . Since  $\mathcal{D}_i \subseteq \mathcal{D}_{i+1}$  and  $|\mathcal{D}_i| = |\mathcal{D}_{i+1}| - 1$ , it follows that  $\mathcal{G}$  contains a clique of size  $|\mathcal{D}_i|$ . It is then easily checked that  $\mathcal{G}$  is a chordal graph.

Since  $\mathcal{G}$  is chordal, we can apply [125, Thm. 7], which states that  $X$  can be completed to a PSD matrix if and only if every fully specified submatrix of  $X$  is PSD. Using similar arguments as in the proof of Lemma 3.25, it follows that every fully specified submatrix of  $X$  is PSD if and only if  $X_{\mathcal{J}}$  is PSD. Recall that  $X_{\mathcal{J}}$  denotes the fully specified principal submatrix of  $X$ , indicated by the elements of

$$\begin{aligned} \mathcal{J} &:= \{\gamma \in \mathcal{D}_{i+1} : X_{2\mathbf{e}_{i+1}, \gamma} \neq \star\} = \{\gamma \in \mathcal{D}_{i+1} : 2\mathbf{e}_{i+1} - \gamma \in \mathcal{D}_i - \mathcal{D}_i\} \\ &= \{\mathbf{e}_{i+1} + \mathbf{e}_k : k \in [n-1]\} \cup \{\mathbf{e}_{i+1}\}. \end{aligned}$$

Note that the linear function  $f(\gamma) := \gamma - \mathbf{e}_{i+1}$  defines a bijection from  $\mathcal{J}$  to a subset of  $\mathcal{D}_i$ . Since  $\gamma - \beta = f(\gamma) - f(\beta)$ , we have that  $(X_{\mathcal{J}})_{\gamma, \beta} = X'_{f(\gamma), f(\beta)}$ . This implies that  $X_{\mathcal{J}}$  is permutation-similar to a submatrix of  $X'$ . Since  $X' \succeq 0$ , also  $X_{\mathcal{J}} \succeq 0$ , which completes the proof. □

*Proof of Lemma 3.46, Page 88.* We show that  $N = G^H G$  satisfies Items 1 and 2 of Proposition 3.45. Denote by  $\mathbf{e}, u, v \in \mathbb{C}^2$  the first three columns of  $G$  (in that order). Observe that  $u_1$  is a convex combination of 1 and  $\exp(2\pi\mathbf{i}/m)$ , both  $m$ -roots of unity, and therefore

$$N_{12} = \mathbf{e}^H u = u_1 = \cos(\pi/m) e^{\pi\mathbf{i}/m} \in \partial\text{Conv}(\mathcal{U}_m) \setminus \mathcal{U}_m.$$

Hence,  $N$  satisfies Item 1 of Proposition 3.45. Let us now verify Item 2 by computing a possible perturbation of  $N$ . Considering (3.72) on Page 86, a perturbation is defined by some  $\alpha \in \mathbb{C}$  and  $c \in \mathbb{R}$ . Since a perturbation (of  $N$ ) remains a perturbation after scaling by a real number, we may assume without loss of generality that  $c = 1$ . We determine  $\alpha$  by solving the linear system (3.71) with  $c = 1$ . The entries of the matrix in this system are given by

$$u_1 \bar{u}_2 = \bar{v}_1 v_2 = \frac{\sin(\frac{2\pi}{m})}{2} e^{\pi\mathbf{i}/m}, |u_2|^2 = \sin^2\left(\frac{\pi}{m}\right) \text{ and } |v_2|^2 = \cos^2\left(\frac{\pi}{m}\right). \quad (\text{B.15})$$

Substituting (B.15) in (3.71) yields

$$\frac{\sin(\frac{2\pi}{m})}{2} \begin{bmatrix} e^{\pi\mathbf{i}/m} & e^{-\pi\mathbf{i}/m} \\ e^{-\pi\mathbf{i}/m} & e^{\pi\mathbf{i}/m} \end{bmatrix} \begin{bmatrix} \alpha \\ \bar{\alpha} \end{bmatrix} = \begin{bmatrix} -\sin^2\left(\frac{\pi}{m}\right) \\ -\cos^2\left(\frac{\pi}{m}\right) \end{bmatrix} = \begin{bmatrix} \frac{\cos(2\pi/m)-1}{2} \\ \frac{-\cos(2\pi/m)-1}{2} \end{bmatrix}. \quad (\text{B.16})$$

Solving (B.16) for the vector  $[\alpha \ \bar{\alpha}]^\top$  yields

$$\begin{aligned} \begin{bmatrix} \alpha \\ \bar{\alpha} \end{bmatrix} &= \frac{1}{\sin(2\pi/m)} \begin{bmatrix} e^{\pi\mathbf{i}/m} & e^{-\pi\mathbf{i}/m} \\ e^{-\pi\mathbf{i}/m} & e^{\pi\mathbf{i}/m} \end{bmatrix}^{-1} \begin{bmatrix} \cos(2\pi/m) - 1 \\ -\cos(2\pi/m) - 1 \end{bmatrix} \\ &= \frac{1}{2\sin^2(2\pi/m)\mathbf{i}} \begin{bmatrix} e^{\pi\mathbf{i}/m} & -e^{-\pi\mathbf{i}/m} \\ -e^{-\pi\mathbf{i}/m} & e^{\pi\mathbf{i}/m} \end{bmatrix} \begin{bmatrix} \cos(2\pi/m) - 1 \\ -\cos(2\pi/m) - 1 \end{bmatrix}. \end{aligned} \quad (\text{B.17})$$

By computing the matrix vector product in (B.17), we obtain

$$\begin{aligned} \alpha &= \frac{-\mathbf{i}}{2\sin^2(\frac{2\pi}{m})} \left( \cos\left(\frac{2\pi}{m}\right) \left[ e^{\frac{\pi\mathbf{i}}{m}} + e^{-\frac{\pi\mathbf{i}}{m}} \right] + e^{-\frac{\pi\mathbf{i}}{m}} - e^{\frac{\pi\mathbf{i}}{m}} \right) \\ &= \frac{-1}{\sin^2(\frac{2\pi}{m})} \left( \cos\left(\frac{2\pi}{m}\right) \cos\left(\frac{\pi}{m}\right) \mathbf{i} + \sin\left(\frac{\pi}{m}\right) \right). \end{aligned}$$

Accordingly,  $b_{12}$ , see (3.72) on Page 86, is computed as follows (using  $c = 1$ ):

$$b_{12} = \mathbf{e}^H \begin{bmatrix} 0 & \bar{\alpha} \\ \alpha & 1 \end{bmatrix} u = u_2 \bar{\alpha}.$$

It remains to show that  $\text{Re}(\bar{\nu} b_{12}) \neq 0$ , for  $\nu = \exp(\pi\mathbf{i}/m)$ . To do so, note that  $u_2 = \sin(\pi/m) \in \mathbb{R}$ , and thus,

$$\begin{aligned} \text{Re}(\bar{\nu} b_{12}) &= \frac{-u_2}{\sin^2(\frac{2\pi}{m})} \text{Re} \left( e^{-\pi\mathbf{i}/m} \left[ \sin\left(\frac{\pi}{m}\right) - \mathbf{i} \cos\left(\frac{\pi}{m}\right) \cos\left(\frac{2\pi}{m}\right) \right] \right) \\ &= \frac{-u_2}{\sin^2(\frac{2\pi}{m})} \cos\left(\frac{\pi}{m}\right) \sin\left(\frac{\pi}{m}\right) \left[ 1 - \cos\left(\frac{2\pi}{m}\right) \right] \neq 0, \text{ since } m > 2. \end{aligned} \quad (\text{B.18})$$

For the second equality in (B.18), we have used that  $\operatorname{Re}(\mathbf{i}e^{-\pi i/m}) = \sin(\pi/m)$ . Since  $b_{12}$  is uniquely determined (up to real scaling), it follows from (B.18) that there does not exist a perturbation satisfying  $\operatorname{Re}(\bar{v}b_{12}) = 0$ . Hence,  $N$  satisfies Item 2 of Proposition 3.45. Thus  $N$  is a rank 2 extreme point of  $\mathcal{E}_m^3$ .  $\square$

### B.3 Proofs from Chapter 4

The following result is used in the proof of Lemma 4.6 on Page 101. This result is also stated in [253], without proof.

**Lemma B.5.** *Let  $n \in \mathbb{N}$  and  $\mathbf{P}^\Pi := \{\bigotimes_{i=1}^n \sigma_i : \sigma_i \in \{\mathbf{I}_2, X, Y, Z\} \forall i \in [n]\}$ , where  $X, Y, Z$  are the  $2 \times 2$  Pauli matrices as in (4.2) on Page 98. Define  $\mathbf{P}_{\mathbb{R}}^\Pi := \mathbf{P}^\Pi \cap \mathbb{R}^{2^n \times 2^n}$  as the set of real matrices in  $\mathbf{P}^\Pi$ . The matrices in  $\mathbf{P}_{\mathbb{R}}^\Pi$  form a basis of  $\mathcal{S}^{2^n}$  over  $\mathbb{R}$ .*

*Proof.* Let  $A = \sigma_1 \otimes \cdots \otimes \sigma_n$  and  $A' = \sigma'_1 \otimes \cdots \otimes \sigma'_n$  be distinct matrices in  $\mathbf{P}^\Pi$ , with  $\sigma_i, \sigma'_i \in \{\mathbf{I}_2, X, Y, Z\}$  for all  $i \in [n]$ . We have that

$$\langle A, A' \rangle = \prod_{i=1}^n \langle \sigma_i, \sigma'_i \rangle = 0,$$

since the matrices in  $\{\mathbf{I}_2, X, Y, Z\}$  are orthogonal with respect to the trace inner product. It follows that the matrices in  $\mathbf{P}^\Pi$  are linearly independent. Therefore also all matrices in  $\mathbf{P}_{\mathbb{R}}^\Pi$  are linearly independent. Because the Kronecker product of Hermitian matrices is Hermitian, it follows that all matrices in  $\mathbf{P}^\Pi$  are Hermitian. Therefore, all matrices in  $\mathbf{P}_{\mathbb{R}}^\Pi$  are real symmetric.

We have thus shown that all matrices in  $\mathbf{P}_{\mathbb{R}}^\Pi$  are linearly independent and real symmetric. It remains to show that the cardinality  $|\mathbf{P}_{\mathbb{R}}^\Pi|$  equals the dimension of  $\mathcal{S}^{2^n}$ , given by  $\dim(\mathcal{S}^{2^n}) = \frac{1}{2}(4^n + 2^n)$ . To compute  $|\mathbf{P}_{\mathbb{R}}^\Pi|$ , note that the Kronecker product of matrices in  $\{\mathbf{I}_2, X, Y, Z\}$  is real if and only if the number of  $Y$  matrices appearing in the product, is even. If  $k$  denotes the number of  $Y$  matrices appearing in such a Kronecker product, then there are  $\binom{n}{k}$  ways to choose the  $k$  positions of  $Y$ . Each of the remaining  $n - k$  positions consists of one of the three matrices in  $\{\mathbf{I}_2, X, Z\}$ . It follows that

$$\begin{aligned} |\mathbf{P}_{\mathbb{R}}^\Pi| &= \sum_{k=0: k \text{ even}}^n \binom{n}{k} 3^{n-k} = \frac{1}{2} \left( \sum_{k=0}^n \binom{n}{k} 3^{n-k} + \sum_{k=0}^n \binom{n}{k} 3^{n-k} (-1)^k \right) \\ &= \frac{1}{2} ((3+1)^n + (3-1)^n) = \dim(\mathcal{S}^{2^n}). \end{aligned} \tag{B.19}$$

The third equality in (B.19) follows from the binomial theorem.  $\square$

We require (a part of) [247, Lem. 1], to prove Lemma B.7, which in turn helps to prove Lemma 4.16 on Page 106.

**Lemma B.6** ([247]). *Let  $L \in F_n^k$  with  $k \geq 2$  and  $n \geq 3$ . Let  $i, j, \ell \in [n]$ , and consider the values  $h_{ij}, h_{i\ell}, h_{j\ell}$  associated to  $L$  as in Definition 4.8. We have that  $h_{ij} + h_{i\ell} + h_{j\ell} \leq 0$ .*

To compare Lemma B.6 with [247, Lem. 1], note that the variables  $s_{ij}$  in [247] satisfy  $s_{ij} = -h_{ij}$ .

**Lemma B.7.** *Let  $G \in \mathbb{G}_s$  with  $s \geq 3$ , and let  $k \geq 2$ . There exists a triangle-free graph  $G' \in \mathbb{G}_s$ , satisfying  $c(G, k) \leq c(G', k)$ , see (4.15).*

*Proof.* If  $G = (V, E)$  is triangle-free, the result follows directly by taking  $G' = G$ . If  $G$  is not triangle-free, we may assume without loss of generality that the edges  $e_1, e_2, e_3 \in E$  form a triangle in  $G$ . Consider an optimal solution of the SDP defining  $c(G, k)$ , with values  $(h_e)_{e \in E}$  as in Definition 4.8. By Lemma B.6,  $\sum_{i=1}^3 h_{e_i} \leq 0$ . Thus, there is some  $e \in \{e_1, e_2, e_3\}$  for which  $h_e \leq 0$ . Consider the graph obtained after removing from  $G$  this edge  $e$ , which is given by  $G[E \setminus e]$ . Observe that  $G[E \setminus e]$  contains strictly less triangles than  $G$ , and satisfies  $c(G, k) \leq c(G[E \setminus e], k)$ . Repeating the procedure if necessary, this concludes the proof.  $\square$

*Proof of Lemma 4.16, Page 106.* The direction  $\Rightarrow$  is trivial, since  $\mathbf{G} \subseteq \mathbb{G}_s$ . For the reverse direction, observe first that  $\max_{G \in \mathbb{G}_s} c(G, k) \geq \lfloor s/2 \rfloor$  (it is straightforward to verify that the graph on  $\lfloor s/2 \rfloor$  disjoint edges, denoted by  $G'$ , satisfies  $c(G', k) = \lfloor s/2 \rfloor$ ). Thus, it remains to prove that all  $G \in \mathbb{G}_s \setminus \mathbf{G}$  satisfy  $c(G, k) \leq \lfloor s/2 \rfloor$ . Due to Lemma B.7 it suffices to consider  $G \in \mathbb{G}_s \setminus \mathbf{G}$  that are triangle-free. We distinguish the following cases, corresponding to which of the four properties of Lemma 4.16 are not satisfied by  $G$ .

**Case 1.** If  $G$  is disconnected, let graphs  $(G^i = (V^i, E^i))_{i \in [p]}$  form the connected components of  $G$  for some  $p \in \mathbb{N}$ . Item 1 from Lemma 4.15 on Page 105 yields

$$c(G, k) \leq \sum_{i=1}^p c(G[E^i], k) \leq \sum_{i=1}^p \left\lfloor \frac{|V^i|}{2} \right\rfloor \leq \left\lfloor \frac{\sum_{i=1}^p |V^i|}{2} \right\rfloor = \left\lfloor \frac{s}{2} \right\rfloor.$$

If  $G$  is connected but not biconnected, there exists a partition  $\{E^1, E^2\}$  of  $E(G)$  such that  $G[E^i]$ ,  $i \in \{1, 2\}$ , is a graph on  $s_i$  vertices, with  $s_1 + s_2 = s + 1$ . Since  $s + 1$  is even, the numbers  $s_i$  are either both even or both odd. If they are both odd, we find

$$c(G, k) \leq \sum_{i=1}^2 c(G[E^i], k) \leq \left\lfloor \frac{s_1}{2} \right\rfloor + \left\lfloor \frac{s_2}{2} \right\rfloor = \frac{s_1 - 1}{2} + \frac{s_2 - 1}{2} = \left\lfloor \frac{s}{2} \right\rfloor.$$

If the numbers  $s_i$  are both even, we proceed as follows: Since  $G$  is connected but not biconnected, the graphs  $G[E^i]$ ,  $i \in \{1, 2\}$ , must have precisely one common vertex  $v \in V(G)$ . Let  $E_v \subseteq E(G)$  be the set of edges adjacent to  $v$ , and consider the partition  $\{E^1 \setminus E_v, E^2 \setminus E_v, E_v\}$  of  $E(G)$ . Observe that  $G[E^i \setminus E_v]$  is a graph on at most  $s_i - 1$  vertices. Also note that  $G[E_v]$  is a star graph, which implies by (4.17) that  $c(G[E_v], k) \leq 1$ . We have

$$c(G, k) \leq c(G[E^1 \setminus E_v], k) + c(G[E^2 \setminus E_v], k) + c(G[E_v], k) \leq \left\lfloor \frac{s}{2} \right\rfloor,$$

where  $c(G[E^i \setminus E_v], k) = 0$ ,  $i \in \{1, 2\}$ , if  $E^i \setminus E_v = \emptyset$ . If  $G$  is bipartite, then its vertex cover number satisfies  $\tau(G) \leq \lfloor s/2 \rfloor$ . Hence, by Item 2 of Lemma 4.15,  $c(G, k) \leq \lfloor s/2 \rfloor$ .

**Case 2.** If  $G$  contains a vertex of degree 1, then  $G$  is not biconnected and the proof follows as in case 1. If  $G$  has a vertex  $i$  satisfying  $\deg(i) > (s - 1)/2$  (equivalently,  $\deg(i) \geq (s + 1)/2$ ), note that the vertices in  $N(i)$  are pairwise non-adjacent since  $G$  is triangle-free. This implies that  $V(G) \setminus N(i)$  is a vertex cover of  $G$ . Then, Item 2 from Lemma 4.15 implies that

$$\begin{aligned} c(G, k) &\leq \tau(G) \leq |V(G) \setminus N(i)| = |V(G)| - |N(i)| = s - \deg(i) \\ &\leq s - (s + 1)/2 = (s - 1)/2 = \lfloor s/2 \rfloor. \end{aligned}$$

**Case 3.** If  $|E(G)| < s$ ,  $G$  is either disconnected or a tree (and thus bipartite). In both cases,  $c(G, k) \leq \lfloor s/2 \rfloor$  (as proven in case 1).

**Case 4.** If  $G = (V, E)$  has a nonempty stable set  $S \subseteq V$  for which  $N(S) := \cup_{i \in S} N(i)$  satisfies  $|N(S)| \leq |S|$ , we proceed as follows: let  $E_{N(S)}$  be the set of edges adjacent to at least one vertex in  $N(S)$ . The set  $N(S)$  is a vertex cover of  $G[E_{N(S)}]$ , so that  $c(G[E_{N(S)}], k) \leq \tau(G[E_{N(S)}]) \leq |N(S)|$  (Item 2 from Lemma 4.15). Observe that  $G[E \setminus E_{N(S)}]$  is a graph on at most  $s - |N(S)| - |S|$  vertices. We find

$$\begin{aligned} c(G, k) &\leq c(G[E_{N(S)}], k) + c(G[E \setminus E_{N(S)}], k) \\ &\leq |N(S)| + \left\lfloor \frac{s - |N(S)| - |S|}{2} \right\rfloor \leq |N(S)| + \left\lfloor \frac{s - 2|N(S)|}{2} \right\rfloor \quad (\text{B.20}) \\ &= |N(S)| + \frac{s - 2|N(S)| - 1}{2} = \frac{s - 1}{2} = \left\lfloor \frac{s}{2} \right\rfloor. \end{aligned}$$

For the third inequality in (B.20), we used that  $|N(S)| \leq |S|$ . For the first equality, we have used that  $s - 2|N(S)|$  is odd, since  $s$  is odd.

We have considered all possible cases of  $G \notin \mathbf{G}$ , which finishes the proof.  $\square$

The following lemma provides two implications of property 4 from Lemma 4.16, Page 106.

**Lemma B.8.** *Let  $G = (V, E)$  be a graph on  $n$  vertices, such that for any stable set  $S$  in  $G$ , it holds that  $|\cup_{i \in S} N(i)| \geq |S| + 1$ . Then  $\deg(i) \geq 2$  for any vertex  $i$ , and the vertex cover number  $\tau(G) \geq (n + 1)/2$ .*

*Proof.* Let  $i \in V$ . Since  $\{i\}$  is a stable set in  $G$ , it holds that  $|N(i)| = \deg(i) \geq |\{i\}| + 1 = 2$ . To prove that  $\tau(G) \geq (n + 1)/2$ , let  $S$  be a stable set of maximum cardinality, i.e.,  $|S| = \alpha(G)$ . As  $S$  is a maximum cardinality stable set, it follows that  $|\cup_{i \in S} N(i)| = n - |S| = n - \alpha(G)$ . Then, by the property of  $G$ ,  $n - \alpha(G) \geq \alpha(G) + 1$ . Substituting the classical result  $\alpha(G) + \tau(G) = n$  [102] completes the proof.  $\square$

The following lemma is used in the proof of Theorem 4.27, Page 114.

**Lemma B.9.** *Let the function  $f$  be defined as in (4.46),  $h$  as in (4.47) and  $r = 0.61392$ . We have that*

$$\max_{p \in \left[ \frac{2r-1}{4(0)-1}, 1 \right]} f(h(r, p), p) < r. \quad (\text{B.21})$$

*Proof.* To simplify notation, we write  $\ell := \frac{2r-1}{q(0)-1}$  and  $g(p) := f(h(r,p),p)$ . We first show that  $g$  is concave. Using concavity, we determine a small interval in which the function  $g$  attains its maximum. We then apply the mean value theorem to this interval, to prove (B.21). For intuition, Figure B.1 provides a plot of  $g$ .

We now prove that  $g$  is concave by showing that  $g''(p) \leq 0$ . Recall from the proof of Theorem 4.27, the expressions of the positive numbers  $\mu = 14/15$ ,  $C_1 = (4 + \mu(\pi - 2))/(2\pi)$  and  $C_2 = (1 + 3\mu/2)$ . The function  $g'(p)$  can be computed by using the multivariate chain rule. To this end, denote by  $D_i f$  the partial derivative of  $f$ , see (4.46), with respect its  $i$ th argument,  $i \in \{1, 2\}$ . For ease of notation, we write  $h(p) = h(r,p)$ , for  $h$  as in (4.47) and  $r = 0.61392$ . We have, for  $p \in (\ell, 1]$ ,

$$\begin{aligned} g'(p) &= h'(p)D_1 f(h(p), p) + D_2 f(h(p), p) \\ &= \frac{(2r-1)q(1/2)}{6pq(0)\sqrt{\frac{2(p+2r-1)}{pq(0)}-1}} \left( \frac{C_1(1-2h(p))}{\sqrt{h(p)(1-h(p))}} - 1 \right) \\ &\quad + \frac{1}{3} \left[ q(1/2) \left( 1 - \frac{h(p)}{2} + C_1 \sqrt{h(p)(1-h(p))} \right) - C_2 \right]. \end{aligned}$$

Observe that  $g$  is not differentiable for  $p = \ell$ , since  $h(\ell) = 0$ . However, it can be verified using a computer that  $r > g(\ell) \approx 0.587$ , so that it suffices to consider only  $p \in (\ell, 1]$ . The function  $g''(p)$  is given by

$$g''(p) = -\frac{1}{3} \left( \frac{2r-1}{q(0)} \right)^2 q(1/2) C_1 \frac{k_1(p)}{k_2(p)},$$

where

$$\begin{aligned} k_1(p) &:= -6u(p) + \frac{2}{C_1} \left( \sqrt{u(p)} - u(p) \right)^{3/2} + 3 \sqrt{u(p)} + 4(u(p))^{3/2}, \\ k_2(p) &:= 4p^3 \left( \sqrt{u(p)} - u(p) \right)^{3/2} (u(p))^{3/2}, \end{aligned}$$

and  $u(p) := \frac{2-q(0)}{q(0)} + \frac{4r-2}{pq(0)}$ . Since  $u(p)$  is decreasing in  $p$  for  $p \in (\ell, 1]$  (and  $u(1) \approx 0.909$ , as can be verified using a computer), we have

$$0 < u(1) \leq u(p) < u(\ell) = 1 \tag{B.22}$$

for all  $p \in (\ell, 1]$ , which implies that  $\sqrt{u(p)} - u(p) > 0$  for  $p \in (\ell, 1]$ . Thus,  $k_1(p)$  and  $k_2(p)$  are well-defined for all  $p \in (\ell, 1]$ . Furthermore, (B.22) shows that  $k_2(p) > 0$  for  $p \in (\ell, 1]$ .

For  $k_1(p)$ , we find  $k_1(p) \geq -6u(p) + 3\sqrt{u(p)} + 4(u(p))^{3/2} \geq 0$ . For the first inequality, we have used that  $\frac{2}{C_1} \left( \sqrt{u(p)} - u(p) \right)^{3/2} \geq 0$ . The second inequality follows from the fact that  $\min_{z \in [0,1]} (-6z + 3\sqrt{z} + 4z^{3/2}) = 0$ , since  $-6z + 3\sqrt{z} + 4z^{3/2}$  is an increasing function, as can be shown by evaluating the derivative.

Hence,  $k_1(p) \geq 0$  and  $k_2(p) > 0$  for all  $p \in (\ell, 1]$ . Since  $-\frac{1}{3} \left( \frac{2r-1}{q(0)} \right)^2 q(1/2) C_1 < 0$ , it follows that  $g''(p) \leq 0$ , which proves that  $g$  is concave on  $p \in (\ell, 1]$ . We will

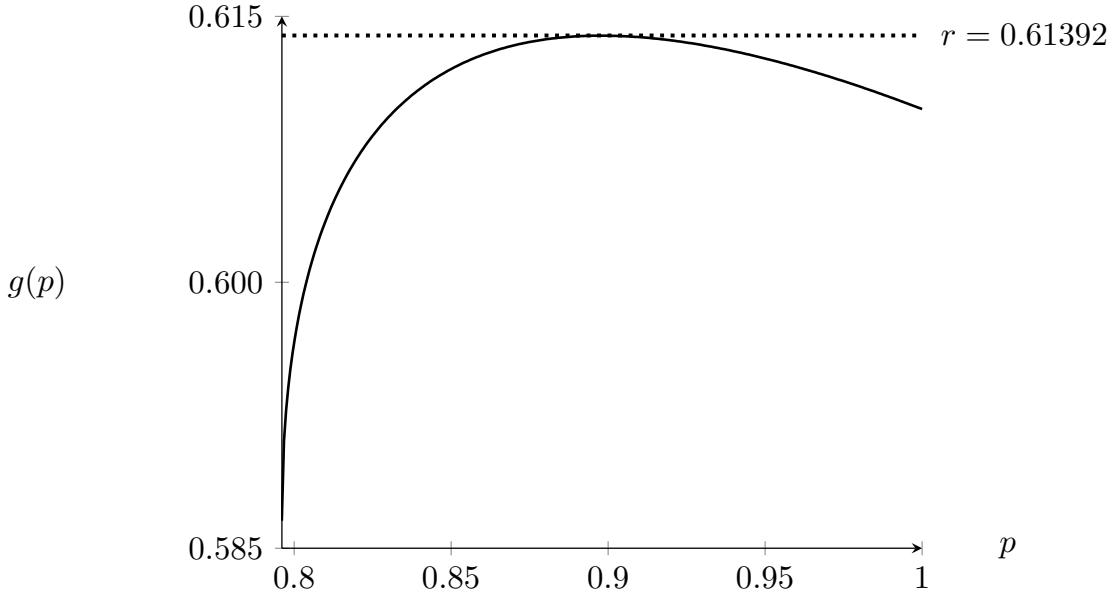


Figure B.1: Plot of the function  $g(p) = f(h(r, p), p)$ , see (B.21), for  $p \in [\ell, 1]$ , where  $\ell = (2r - 1)/(q(0) - 1) \approx 0.7961$ .

use concavity of  $g$ , in combination with the mean value theorem, to prove (B.21). Let  $p_1 = 0.897$  and  $p_2 = 0.898$ . It can be verified (by computer) that  $g'(p_1) > 0$  and  $g'(p_2) < 0$ . By concavity of  $g$ , it follows that  $\max_{p \in [\ell, 1]} g(p) = g(p^*)$ , for some  $p^* \in (p_1, p_2)$ . Observe that  $g$  is continuous and continuously differentiable on  $(p_1, p_2)$ . By the mean value theorem, we have

$$\max_{p \in [\ell, 1]} g(p) = g(p^*) = g'(z)(p^* - p_1) + g(p_1) \leq g'(p_1)(p_2 - p_1) + g(p_1) < r,$$

for some  $z \in (p_1, p_2)$ . For the first inequality, we have used that  $g$  is concave, so that  $g'(p)$  is a decreasing function. The second inequality can be verified by computing the number  $g'(p_1)(p_2 - p_1) + g(p_1) (\approx 0.61391)$ .  $\square$

**Lemma B.10.** *The system of inequalities (4.77) is inconsistent for  $r := 0.8339$ .*

*Proof.* Define

$$\begin{aligned} A_1 &:= \{z \in [0, 1]^2 : 2r - 1 \leq (1 - z_1)(1 - z_2), 0 \leq z_1 \leq z_2 \leq 2(1 - r)\} \\ A_2 &:= \{z \in [0, 1]^2 : 2\sqrt{z_1(1 - z_2)} - z_2 \geq c_2, 0 \leq z_1 \leq z_2 \leq 2(1 - r)\} \\ A_3 &:= \{z \in [0, 1]^2 : 2\sqrt{z_2(1 - z_1)} - z_1 \geq c_3, 0 \leq z_1 \leq z_2 \leq 2(1 - r)\}, \end{aligned}$$

where  $c_2 := (4 - \sqrt{3})r - 2 \approx -0.109$ , and  $c_3 := (2 + \sqrt{3})r - 2 \approx 1.112$ . If  $z$  is a solution to (4.77), then  $z \in A := A_1 \cap A_2 \cap A_3$ . We will show that  $A = \emptyset$ , which implies that (4.77) is inconsistent. Proving  $A = \emptyset$  directly is difficult, due to the nonlinearity of  $A$ . To circumvent this difficulty, we will define half-planes  $H_i$ ,  $i \in \{1, 2, 3\}$ , that satisfy  $A_i \subseteq H_i$ . Therefore, the intersection  $H := H_1 \cap H_2 \cap H_3 \cap \{z \in \mathbb{R}^2 : z \geq 0\}$

satisfies  $A \subseteq H$ , so that  $A = \emptyset$  follows from  $H = \emptyset$ . Since  $H$  is a polytope,  $H = \emptyset$  can be proven via Farkas' lemma [85].

Let us now define the half-planes  $H_i$  for  $i \in \{1, 2, 3\}$ . We also define corresponding functions  $f_i$ , that we use to prove the inclusions  $A_i \subseteq H_i$ . Consider

$$\begin{aligned} H_1 &:= \{z \in \mathbb{R}^2 : 0.69z_1 + z_2 \leq 0.333\}, & f_1(x) &:= 0.69x + \frac{2r + x - 2}{x - 1}, \\ H_2 &:= \{z \in \mathbb{R}^2 : z_1 - 0.183z_2 \geq -0.0423\}, & f_2(x) &:= \frac{\max\{c_2 + x, 0\}^2}{4(1-x)} - 0.183x, \\ H_3 &:= \{z \in \mathbb{R}^2 : -0.9z_1 + z_2 \geq 0.3088\}, & f_3(x) &:= -0.9x + \frac{(c_3 + x)^2}{4(1-x)}. \end{aligned}$$

To prove that  $A_1 \subseteq H_1$ , observe first that  $z \in A_1 \implies z_2 \leq \frac{2r+z_1-2}{z_1-1}$  and  $z_1 \in [0, 2(1-r)]$ . For  $z \in A_1$ , we have

$$0.69z_1 + z_2 \leq 0.69z_1 + \frac{2r + z_1 - 2}{z_1 - 1} \leq \max_{x \in [0, 2(1-r)]} f_1(x) = f_1(x^*) < 0.333,$$

where  $x^* (\approx 0.016)$  is the stationary point of  $f_1$  in the interval  $[0, 2(1-r)]$  (we omit the computation of  $x^*$ ), and  $f_1(x^*) \approx 0.332$ . Thus,  $A_1 \subseteq H_1$ .

For  $A_2 \subseteq H_2$ , we use that  $z \in A_2 \implies z_1 \geq \frac{\max\{c_2+z_2, 0\}^2}{4(1-z_2)}$  and  $z_2 \in [0, 2(1-r)]$ . Then

$$z_1 - 0.183z_2 \geq \frac{\max\{c_2 + z_2, 0\}^2}{4(1-z_2)} - 0.183z_2 \geq \min_{x \in [0, 2(1-r)]} f_2(x). \quad (\text{B.23})$$

To solve the minimization problem in (B.23), note that  $f_2(x) = -0.183x$  for  $x \in [0, -c_2]$ . Thus,  $\min_{x \in [0, -c_2]} f_2(x) = 0.183c_2 \approx -0.020$ . For  $x \in [-c_2, 2(1-r)]$ , we have  $\min_{x \in [-c_2, 2(1-r)]} f_2(x) = f_2(x^*) \approx -0.0422$ . Here,  $x^* (\approx 0.323)$  is the stationary point of  $f_2$  in  $[-c_2, 2(1-r)]$  (we omit the computation of  $x^*$ ). We conclude that  $\min_{x \in [0, 2(1-r)]} f_2(x) = f_2(x^*) > -0.0423$ , which implies by (B.23) that  $A_2 \subseteq H_2$ .

The proof of  $A_3 \subseteq H_3$  is similar to the proof of  $A_2 \subseteq H_2$ . We have  $z \in A_3 \implies z_2 \geq \frac{(c_3+z_1)^2}{4(1-z_1)}$  and  $z_1 \in [0, 2(1-r)]$ . Then

$$-0.9z_1 + z_2 \geq -0.9z_1 + \frac{(c_3 + z_1)^2}{4(1-z_1)} \geq \min_{x \in [0, 2(1-r)]} f_3(x) = f_3(x^*) > 0.3088,$$

where  $x^* (\approx 0.015)$  is the stationary point of  $f_3$  in  $[0, 2(1-r)]$ , and  $f_3(x^*) \approx 0.309$ . Thus,  $A_i \subseteq H_i$  for all  $i \in \{1, 2, 3\}$ .

Using slack variables  $s$ , we can write the intersection  $H = H_1 \cap H_2 \cap H_3$  as follows:  $H$  consists of all the  $z \in \mathbb{R}^2$ ,  $z \geq 0$ , for which there exists an  $s \in \mathbb{R}^3$ ,  $s \geq 0$ , such that  $M [z^\top, s^\top]^\top = u$ , where

$$M := \begin{bmatrix} 0.69 & 1 & 1 & 0 & 0 \\ 1 & -0.183 & 0 & -1 & 0 \\ -0.9 & 1 & 0 & 0 & -1 \end{bmatrix} \quad \text{and} \quad u := \begin{bmatrix} 0.333 \\ -0.0423 \\ 0.3088 \end{bmatrix}.$$

By Farkas' lemma [85],  $H$  is empty if and only if there exists a vector  $y \in \mathbb{R}^3$  satisfying  $M^\top y \geq 0$  and  $u^\top y < 0$ . Since  $y := [0.824, -1.447, -1.087]^\top$  satisfies these conditions,  $H = \emptyset$ , which implies that  $A = \emptyset$  because  $A \subseteq H$ . Since any solution to (4.77) is contained in  $A$ , it follows that (4.77) is inconsistent.  $\square$

# Academic summary

A mathematical optimization problem, also referred to as a program, is defined by an objective function and a set of constraint functions. Solving such a program consists of assigning values to the variables of these functions, in such a way that the objective function is maximized, or minimized, and the constraints are satisfied. Mathematical optimization is widely used in many fields of science, in business and in engineering. For example, mathematical optimization can be used to determine an asset portfolio that maximizes the expected return, while also respecting constraints involving risk measures.

Arguably, the simplest programs are those in which the objective and constraint functions are linear functions. Such linear programs (LPs) are well understood, and widely used in practice. A slightly more general program is known as a semidefinite program (SDP). SDPs are programs with a linear objective function, and constraints that require that certain matrix-valued variables are positive semidefinite. Both SDPs and LPs can be solved efficiently. Roughly speaking, the number of steps required to solve LPs and SDPs is polynomial in the input size of the program.

This thesis studies SDPs, methods for solving them, and their use for solving stable set and max-cut problems. The stable set and max-cut problems are fundamental problems from computer science, with many applications. These two problems generally cannot be solved efficiently, unlike LPs and SDPs. As large scale problem instances of the stable set and max-cut problems are thus practically unsolvable, one often resorts to so-called relaxations of these problems that can be solved efficiently. These relaxations provide approximate solutions to the original problem. Relaxations based on SDP, i.e., SDP relaxations, are more accurate than those based on LP, but also require more computation time.

This thesis considers SDP relaxations for (variants of) the stable set and max-cut problems. These relaxations are usually solved by computers and do not admit closed form solutions. However, we also consider certain problem instances for which the SDP relaxations provide closed form solutions. Another point of study in this thesis is which SDP relaxation to use: there exist hierarchies of SDP relaxations of increasing accuracy, and increasing difficulty in solving them. The goal is to select the relaxation that is easiest to solve, while also being sufficiently accurate. To do so, it is important to understand how the levels of the hierarchy differ.

SDP relaxations are often used in so-called approximation algorithms. These are efficient algorithms that provide an approximate solution to some problem, with provable performance guarantees on the accuracy of the approximate solution. This

performance guarantee is referred to as the approximation ratio of the approximation algorithm. For a variant of the max-cut problem known as the quantum max-cut (QMC) problem, it is unclear what the best possible approximation ratio is. This thesis studies approximation algorithms for the QMC problem that use SDP relaxations. In particular, this thesis provides improved approximation ratios and approximation algorithms on certain QMC problem classes.

The previously discussed topics concern theoretical properties of (algorithms that involve) SDPs. In this thesis we also investigate computation methods for solving SDPs. There exist various computation methods for solving SDPs, each well suited to different purposes. In this thesis, we solve large stable set problems using SDP relaxations that we solve with a particular computation method. We show that this computation method works well via extensive numerical benchmarks. In particular, it is shown that this method requires significantly less computer memory than other methods. This ensures that the large scale relaxations can be solved on modest computer hardware, without running out of memory.

We use a similar computation method for solving SDP relaxations of the MAX-SAT problem, which is similar to the max-cut problem. We provide a MAX-SAT solver that uses these SDP relaxations. This algorithm is compared to other MAX-SAT solvers from the literature. This comparison highlights the strength of SDP relaxations for solving certain MAX-SAT problems.

# Academische samenvatting

Een wiskundig optimalisatieprobleem, ook wel een programma genoemd, bestaat uit een doelfunctie en een verzameling van randvoorwaarden. Het oplossen van een dergelijk programma bestaat uit het toekennen van waarden aan de variabelen van deze functies, zodanig dat de doelfunctie wordt gemaximaliseerd of geminimaliseerd en aan de randvoorwaarden wordt voldaan. Wiskundige optimalisatie wordt op grote schaal toegepast in vele wetenschapsgebieden, in het bedrijfsleven en in de techniek. Zo kan wiskundige optimalisatie worden gebruikt om een beleggingsportefeuille te bepalen die het verwachte rendement maximaliseert, terwijl tegelijkertijd wordt voldaan aan randvoorwaarden die betrekking hebben op risicomaten.

De eenvoudigste programma's zijn waarschijnlijk die waarbij de doel- en randvoorwaarden lineaire functies zijn. Dergelijke lineaire programma's (LP's) zijn goed begrepen en worden in de praktijk veelvuldig gebruikt. Een iets algemener programma staat bekend als een semidefiniet programma (SDP). SDP's zijn programma's met een lineaire doelfunctie en randvoorwaarden die vereisen dat bepaalde matrix variabelen positief semidefiniet zijn. Zowel SDP's als LP's kunnen efficiënt worden opgelost: grofweg gesteld is het aantal stappen dat nodig is om LP's en SDP's op te lossen polynomiaal in de grootte van het programma.

Dit proefschrift bestudeert SDP's, methoden om deze op te lossen, en hun toepassing bij het oplossen van het stabiele verzameling probleem en het max-cut probleem. Het stabiele verzameling probleem en het max-cut probleem zijn fundamentele problemen uit de informatica, met vele toepassingen. Deze twee problemen kunnen in het algemeen niet efficiënt worden opgelost, in tegenstelling tot LP's en SDP's. Omdat grootschalige probleeminstanties van het stabiele verzameling en max-cut probleem daardoor in de praktijk onoplosbaar zijn, wordt vaak gebruikgemaakt van zogenoemde relaxaties van deze problemen die wel efficiënt kunnen worden opgelost. Deze relaxaties leveren benaderende oplossingen voor het oorspronkelijke probleem. Op SDP gebaseerde relaxaties, dat wil zeggen SDP relaxaties, zijn nauwkeuriger dan relaxaties gebaseerd op LP, maar vereisen ook meer rekentijd.

Dit proefschrift beschouwt SDP relaxaties voor (varianten van) het stabiele verzameling en het max-cut probleem. Deze relaxaties worden doorgaans door computers opgelost en laten geen gesloten-vormoplossingen toe. We beschouwen ook bepaalde probleeminstanties waarvoor de SDP relaxaties wel gesloten-vormoplossingen opleveren. Een ander aandachtspunt in dit proefschrift is de keuze van de SDP relaxatie: er bestaan hiërarchieën van SDP relaxaties met toenemende nauwkeurigheid en toenemende moeilijkheidsgraad om ze op te lossen. Het doel is om de relaxatie te selecteren

die het eenvoudigst op te lossen is en tegelijkertijd voldoende nauwkeurig is. Om dit te kunnen doen, is het belangrijk te begrijpen waarin de niveaus van de hiërarchie van elkaar verschillen.

SDP relaxaties worden vaak gebruikt in zogenoemde benaderingsalgoritmen. Dit zijn efficiënte algoritmen die een benaderende oplossing voor een probleem leveren, met bewijsbare prestatiegaranties voor de nauwkeurigheid van die oplossing. Deze prestatiegarantie wordt de benaderingsratio van het benaderingsalgoritme genoemd. Voor een variant van het max-cut probleem, bekend als het quantum max-cut probleem (QMC), is het onduidelijk wat de best mogelijke benaderingsratio is. Dit proefschrift bestudeert benaderingsalgoritmen voor het QMC probleem die gebruikmaken van SDP relaxaties. In het bijzonder presenteert dit proefschrift verbeterde benaderingsratio's en benaderingsalgoritmen voor bepaalde klassen van QMC problemen.

De eerder besproken onderwerpen betreffen theoretische eigenschappen van (algoritmen die gebruikmaken van) SDP's. In dit proefschrift onderzoeken we ook numerieke methoden voor het oplossen van SDP's. Er bestaan verschillende numerieke methoden voor het oplossen van SDP's, die elk geschikt zijn voor verschillende doeleinden. In dit proefschrift lossen we grote stabiele verzameling problemen op met behulp van SDP relaxaties, die we oplossen met een specifieke numerieke methode. Aan de hand van uitgebreide numerieke experimenten tonen we aan dat deze methode goed werkt. In het bijzonder wordt aangetoond dat deze methode aanzienlijk minder computergeheugen vereist dan andere methoden. Dit maakt het mogelijk om grootschalige relaxaties op bescheiden computerhardware op te lossen zonder dat het geheugen uitgeput raakt.

We gebruiken een vergelijkbare numerieke methode voor het oplossen van SDP relaxaties van het MAX-SAT probleem, dat vergelijkbaar is met het max-cut probleem. We presenteren een MAX-SAT algoritme die gebruikmaakt van deze SDP relaxaties. Dit algoritme wordt vergeleken met andere MAX-SAT algoritmes uit de literatuur. Deze vergelijking benadrukt de kracht van SDP relaxaties voor het oplossen van bepaalde MAX-SAT problemen.

# Bibliography

- [1] H. Abdo and D. Dimitrov. The total irregularity of graphs under graph operations. *Miskolc Mathematical Notes*, 15(1):3–17, 2014.
- [2] A. Abramé and D. Habet. AHMAXSAT: Description and evaluation of a branch and bound Max-SAT solver. *Journal on Satisfiability, Boolean Modeling and Computation*, 9(1):89–128, 2014.
- [3] E. Adams, M. F. Anjos, F. Rendl, and A. Wiegele. A hierarchy of subgraph projection-based semidefinite relaxations for some NP-hard graph optimization problems. *Information Systems and Operational Research*, 53(1):40–48, 2015.
- [4] L. Addario-Berry, W. Kennedy, A. D. King, Z. Li, and B. Reed. Finding a maximum-weight induced  $k$ -partite subgraph of an  $i$ -triangulated graph. *Discrete Applied Mathematics*, 158(7):765–770, 2010.
- [5] A. A. Ahmadi, G. Hall, A. Papachristodoulou, J. Saunderson, and Y. Zheng. Improving efficiency and scalability of sum of squares optimization: Recent advances and limitations. In *2017 IEEE 56th annual conference on decision and control (CDC)*, pages 453–462. IEEE, 2017.
- [6] W. Ai, Y. Huang, and S. Zhang. New results on Hermitian matrix rank-one decomposition. *Mathematical programming*, 128(1-2):253–283, 2011.
- [7] F. Alizadeh. Interior point methods in semidefinite programming with applications to combinatorial optimization. *SIAM Journal on Optimization*, 5(1):13–51, 1995.
- [8] M. F. Anjos. *New Convex Relaxations for the Maximum Cut and VLSI Layout Problems*. PhD thesis, University of Waterloo, 2001.
- [9] M. F. Anjos. On semidefinite programming relaxations for the satisfiability problem. *Mathematical Methods of Operations Research*, 60(3):349–367, 2004.
- [10] M. F. Anjos. An improved semidefinite programming relaxation for the satisfiability problem. *Mathematical Programming*, 102(3):589–608, 2005.
- [11] M. F. Anjos. Semidefinite optimization approaches for satisfiability and maximum-satisfiability problems. *Journal on Satisfiability, Boolean Modeling and Computation*, 1(1):1–47, 2006.
- [12] M. F. Anjos. An extended semidefinite relaxation for satisfiability. *Journal on Satisfiability, Boolean Modeling and Computation*, 4(1):15–31, 2007.
- [13] M. F. Anjos and J. B. Lasserre, editors. *Handbook on semidefinite, conic and polynomial optimization*. Springer Science & Business Media, 2011.
- [14] M. F. Anjos and H. Wolkowicz. Geometry of semidefinite Max-Cut relaxations

- via matrix ranks. *Journal of Combinatorial Optimization*, 6(3):237–270, 2002.
- [15] A. Anshu, D. Gosset, and K. Morenz. Beyond product state approximations for a quantum analogue of max cut. In S. T. Flammia, editor, *15th Conference on the Theory of Quantum Computation, Communication and Cryptography*, volume 158 of *LIPICs*, pages 7:1–7:15, 2020.
- [16] M. Aouchiche and P. Hansen. A survey of Nordhaus–Gaddum type relations. *Discrete Applied Mathematics*, 161(4-5):466–546, 2013.
- [17] K. Appel and W. Haken. Every planar map is four colorable. Part I: Discharging. *Illinois Journal of Mathematics*, 21(3):429–490, 1977.
- [18] A. Apte, E. Lee, K. Marwaha, O. Parekh, and J. Sud. Improved algorithms for quantum maxcut via partially entangled matchings. *preprint arXiv:2504.15276*, 2025.
- [19] R. Asín Achá and R. Nieuwenhuis. Curriculum-based course timetabling with SAT and MaxSAT. *Annals of Operations Research*, 218(1):71–91, 2014.
- [20] B. Aspvall, M. F. Plass, and R. E. Tarjan. A linear-time algorithm for testing the truth of certain quantified boolean formulas. *Information processing letters*, 8(3):121–123, 1979.
- [21] C. Bachoc, A. Pêcher, and A. Thiéry. On the theta number of powers of cycle graphs. *Combinatorica*, 33(3):297–317, 2013.
- [22] W. N. Bailey. Generalized hypergeometric series. *Cambridge Tracts in Mathematics and Mathematical Physics*, 1935.
- [23] E. Balas and J. Xue. Weighted and unweighted maximum clique algorithms with upper bounds from fractional coloring. *Algorithmica*, 15(5):397–412, 1996.
- [24] A. S. Bandeira, N. Boumal, and A. Singer. Tightness of the maximum likelihood semidefinite relaxation for angular synchronization. *Mathematical Programming*, 163:145–167, 2017.
- [25] F. Barahona and A. R. Mahjoub. On the cut polytope. *Mathematical programming*, 36:157–173, 1986.
- [26] F. Barahona, M. Grötschel, M. Jünger, and G. Reinelt. An application of combinatorial optimization to statistical physics and circuit layout design. *Operations Research*, 36(3):493–513, 1988.
- [27] Z. Baranyai. On the factorization of the complete uniform hypergraphs. In R. R. A. Hajnal and V. Sós, editors, *Infinite and finite sets, Proc. Intern. Coll. Keszthely*, pages 91–108. North-Holland, Amsterdam, 1973.
- [28] A. Ben-Tal and A. Nemirovski. On polyhedral approximations of the second-order cone. *Mathematics of Operations Research*, 26(2):193–205, 2001.
- [29] G. Blekherman, P. A. Parrilo, and R. R. Thomas, editors. *Semidefinite optimization and convex algebraic geometry*. SIAM, 2012.
- [30] N. Blum. A new approach to maximum matching in general graphs. In *Proceedings of the 47th International Colloquium on Automata, Languages and Programming*, pages 586–597, 1990.
- [31] F. Boesch and R. Tindell. Circulants and their connectivities. *Journal of Graph Theory*, 8(4):487–499, 1984.
- [32] M. L. Bonet, J. Levy, and F. Manyá. Resolution for Max-SAT. *Artificial*

- Intelligence*, 171(8-9):606–618, 2007.
- [33] S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge university press, 2004.
- [34] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine learning*, 3(1):1–122, 2011.
- [35] J. Brakensiek, N. Huang, and U. Zwick. Tight approximability of MAX 2-SAT and relatives, under UGC. In *Proceedings of the 2024 Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1328–1344. SIAM, 2024.
- [36] G. Braun and S. Pokutta. The matching problem has no fully polynomial size linear programming relaxation schemes. *IEEE Transactions on Information Theory*, 61(10):5754–5764, 2015.
- [37] J. Briët, F. M. de Oliveira Filho, and F. Vallentin. Grothendieck inequalities for semidefinite programs with rank constraint. *Theory of Computing*, 10(4):77–105, 2014.
- [38] R. C. Brigham and R. D. Dutton. Generalized  $k$ -tuple colorings of cycles and other graphs. *Journal of Combinatorial Theory, Series B*, 32(1):90–94, 1982.
- [39] V. E. Brimkov. Algorithmic and explicit determination of the Lovász number for certain circulant graphs. *Discrete Applied Mathematics*, 155(14):1812–1825, 2007.
- [40] V. E. Brimkov, B. Codenotti, V. Crespi, and M. Leoncini. On the Lovász number of certain circulant graphs. In G. C. Bongiovanni, G. Gambosi, and R. Petreschi, editors, *Algorithms and Complexity, Proceedings*, volume 1767 of *Lecture Notes in Computer Science*, pages 291–305. Springer, 2000.
- [41] G. Brinkmann, K. Coolsaet, J. Goedgebeur, and H. Mélot. House of graphs: a database of interesting graphs. *Discrete Applied Mathematics*, 161(1-2):311–314, 2013.
- [42] A. E. Brouwer and W. H. Haemers. *Spectra of graphs*. Springer Science & Business Media, 2011.
- [43] A. E. Brouwer, A. M. Cohen, and A. Neumaier. *Distance-regular graphs*. Berlin; New York: Springer-Verlag, 1989.
- [44] A. E. Brouwer, S. M. Cioabă, F. Ihringer, and M. McGinnis. The smallest eigenvalues of Hamming graphs, Johnson graphs and other distance-regular graphs with classical parameters. *Journal of Combinatorial Theory, Series B*, 133:88–121, 2018.
- [45] S. Burgdorf, I. Klep, and J. Povh. *Optimization of polynomials in non-commuting variables*. SpringerBriefs in Mathematics. Springer, 2016.
- [46] F. Burkowski, H. Im, and H. Wolkowicz. A Peaceman-Rachford splitting method for the protein side-chain positioning problem. *INFORMS Journal on Computing*, 2024.
- [47] M. Campêlo, R. C. Corrêa, P. F. Moura, and M. C. Santos. On optimal  $k$ -fold colorings of webs and antiwebs. *Discrete Applied Mathematics*, 161(1-2):60–70, 2013.
- [48] M. Campêlo, P. F. Moura, and M. C. Santos. Lifted, projected and subgraph-induced inequalities for the representatives  $k$ -fold coloring polytope. *Discrete*

- Optimization*, 21:131–156, 2016.
- [49] J. S. Campos, R. Misener, and P. Parpas. Partial Lasserre relaxation for sparse max-cut. *Optimization and Engineering*, 24:1983–2004, 2023.
- [50] G. J. Chaitin. Register allocation and spilling via graph coloring. *ACM SIGPLAN Notices*, 17(6):98–101, 1982.
- [51] T. H. Chan, K. L. Chang, and R. Raman. An SDP primal-dual algorithm for approximating the Lovász-theta function. In *2009 IEEE International Symposium on Information Theory, Korea (South)*, pages 2808–2812. IEEE, 2009.
- [52] L.-C. Chang. The uniqueness and nonuniqueness of the triangular association scheme. *Science Record*, 3:604–613, 1959.
- [53] B.-L. Chen and K.-W. Lih. Hamiltonian uniform subset graphs. *Journal of Combinatorial Theory, Series B*, 42(3):257–263, 1987.
- [54] T. Chen, J. B. Lasserre, V. Magron, and E. Pauwels. A sublevel moment-SOS hierarchy for polynomial optimization. *Computational Optimization and Applications*, 81(1):31–66, 2022.
- [55] S. Chopra and M. R. Rao. Facets of the  $k$ -partition polytope. *Discrete Applied Mathematics*, 61(1):27–48, 1995.
- [56] J. P. R. Christensen and J. Vesterstrøm. A note on extreme positive definite matrices. *Mathematische Annalen*, 244:65–68, 1979.
- [57] M. Chudnovsky, N. Robertson, P. Seymour, and R. Thomas. The strong perfect graph theorem. *Annals of mathematics*, pages 51–229, 2006.
- [58] V. Chvátal. Edmonds polytopes and a hierarchy of combinatorial problems. *Discrete Mathematics*, 4(4):305–337, 1973.
- [59] C. Coey, L. Kapelevich, and J. P. Vielma. Solving natural conic formulations with Hypatia.jl. *INFORMS Journal on Computing*, 34(5):2686–2699, 2022.
- [60] S. A. Cook. The complexity of theorem-proving procedures. In *Proceedings of the third annual ACM Symposium on Theory of Computing*, pages 151–158, 1971.
- [61] D. W. Cranston and L. Rabern. Planar graphs are  $9/2$ -colorable. *Journal of Combinatorial Theory, Series B*, 133:32–45, 2018.
- [62] V. Crespi. Exact formulae for the Lovász theta function of sparse circulant graphs. *SIAM Journal on Discrete Mathematics*, 17(4):670–674, 2004.
- [63] T. Cubitt and A. Montanaro. Complexity classification of local Hamiltonian problems. *SIAM Journal on Computing*, 45(2):268–316, 2016.
- [64] R. E. Curto and L. A. Fialkow. *Flat Extensions of Positive Moment Matrices: Recursively Generated Relations*, volume 136 of *Memoirs of the American Mathematical Society*. American Mathematical Soc., 1998.
- [65] E. de Klerk. Exploiting special structure in semidefinite programming: A survey of theory and applications. *European Journal of Operational Research*, 201:1–10, 2010.
- [66] E. de Klerk and M. Laurent. On the Lasserre hierarchy of semidefinite programming relaxations of convex polynomial optimization problems. *SIAM Journal on Optimization*, 21(3):824–832, 2011.
- [67] E. de Klerk and D. V. Pasechnik. A note on the stability number of an orthog-

- onality graph. *European Journal of Combinatorics*, 28(7):1971–1979, 2007.
- [68] E. de Klerk, H. van Maaren, and J. Warners. Relaxations of the satisfiability problem using semidefinite programming. *Journal of automated reasoning*, 24(1):37–65, 2000.
- [69] F. de Meijer and R. Sotirov. SDP-based bounds for the quadratic cycle cover problem via cutting-plane augmented Lagrangian methods and reinforcement learning. *INFORMS Journal on Computing*, 33(4):1262–1276, 2021.
- [70] F. de Meijer, R. Sotirov, A. Wiegele, and S. Zhao. Partitioning through projections: Strong SDP bounds for large graph partition problems. *Computers & Operations Research*, 151:106088, 2023.
- [71] C. Delorme and S. Poljak. Combinatorial properties and the complexity of a max-cut approximation. *European Journal of Combinatorics*, 14(4):313–333, 1993.
- [72] C. Delorme and S. Poljak. Laplacian eigenvalues and the maximum cut problem. *Mathematical Programming*, 62:557–574, 1993.
- [73] P. Delsarte. Hahn polynomials, discrete harmonics, and  $t$ -designs. *SIAM Journal on Applied Mathematics*, 34(1):157–166, 1978.
- [74] M. M. Deza and M. Laurent. *Geometry of cuts and metrics*. Springer, 1997.
- [75] C. H. Ding, X. He, H. Zha, M. Gu, and H. D. Simon. A min-max cut algorithm for graph partitioning and data clustering. In *Proceedings 2001 IEEE international conference on data mining*, pages 107–114. IEEE, 2001.
- [76] C. Dobre. *Semidefinite programming approaches for structured combinatorial optimization problems*. PhD thesis, Tilburg University, 2011.
- [77] D. Drusvyatskiy, G. Li, and H. Wolkowicz. A note on alternating projections for ill-posed semidefinite feasibility problems. *Mathematical Programming*, 162(1):537–548, 2017.
- [78] I. Dukanovic and F. Rendl. Semidefinite programming relaxations for graph coloring and maximal clique problems. *Mathematical Programming*, 109(2-3):345–365, 2007.
- [79] N. Dyn and W. E. Ferguson. The numerical solution of equality constrained quadratic programming problems. *Mathematics of Computation*, 41(163):165–170, 1983.
- [80] J. Edmonds. Maximum matching and a polyhedron with 0, 1-vertices. *Journal of Research of the National Bureau of Standards, Section B: Mathematics and Mathematical Physics*, 69B:125–130, 1965.
- [81] J. Edmonds. Paths, trees, and flowers. *Canadian Journal of Mathematics*, 17:449–467, 1965.
- [82] S. Y. El Rouayheb, C. N. Georghiades, E. Soljanin, and A. Sprintson. Bounds on codes based on graph theory. In *2007 IEEE International Symposium on Information Theory, Nice, France*, pages 1876–1879. IEEE, 2007.
- [83] K. Fan. On a theorem of Weyl concerning eigenvalues of linear transformations i. *Proceedings of the National Academy of Sciences*, 35(11):652–655, 1949.
- [84] E. Farhi, J. Goldstone, and S. Gutmann. A quantum approximate optimization algorithm. *preprint arXiv:1411.4028*, 2014.

- [85] J. Farkas. Theorie der einfachen ungleichungen. *Journal für die reine und angewandte Mathematik (Crelles Journal)*, 124:1–27, 1902.
- [86] H. Fawzi, J. Saunderson, and P. A. Parrilo. Sparse sums of squares on finite abelian groups and improved semidefinite lifts. *Mathematical Programming*, 160:149–191, 2016.
- [87] H. Fawzi, J. Saunderson, and P. A. Parrilo. Equivariant semidefinite lifts of regular polygons. *Mathematics of Operations Research*, 42(2):472–494, 2017.
- [88] U. Feige and M. X. Goemans. Approximating the value of two power proof systems, with applications to MAX 2SAT and MAX DICUT. In *Proceedings third Israel symposium on the theory of computing and systems*, pages 182–189. IEEE, 1995.
- [89] S. Fiorini, T. Rothvoß, and H. R. Tiwary. Extended formulations for polygons. *Discrete & computational geometry*, 48(3):658–668, 2012.
- [90] M. Fortin and R. Glowinski. On decomposition-coordination methods using an augmented Lagrangian. In M. Fortin and R. Glowinski, editors, *Augmented Lagrangian Methods: Applications to the Numerical Solution of Boundary-Value Problems*, volume 15, pages 97–146. Elsevier, 1983.
- [91] P. Fouilhoux and A. R. Mahjoub. Solving VLSI design and DNA sequencing problems using bipartization of graphs. *Computational Optimization and Applications*, 51(2):749–781, 2012.
- [92] P. Frankl. Orthogonal vectors in the  $n$ -dimensional cube and codes with missing distances. *Combinatorica*, 6:279–285, 1986.
- [93] P. Frankl and V. Rödl. Forbidden intersections. *Trans. Amer. Math. Soc.*, 300: 259–286, 1987.
- [94] P. Frankl and N. Tokushige. The Erdős–Ko–Rado theorem for integer sequences. *Combinatorica*, 19(1):55–63, 1999.
- [95] A. Frieze and M. Jerrum. Improved approximation algorithms for MAX  $k$ -CUT and MAX BISECTION. *Algorithmica*, 18(1):67–81, 1997.
- [96] Z. Füredi and P. Frankl. Extremal problems concerning Kneser graphs. *Journal of Combinatorial Theory, Series B*, 40(3):270–284, 1986.
- [97] E. Gaar. On different versions of the exact subgraph hierarchy for the stable set problem. *Discrete Applied Mathematics*, 356:52–70, 2024.
- [98] E. Gaar and F. Rendl. A computational study of exact subgraph based SDP bounds for max-cut, stable set and coloring. *Mathematical Programming*, 183 (1):283–308, 2020.
- [99] E. Gaar, M. Siebenhofer, and A. Wiegele. An SDP-based approach for computing the stability number of a graph. *Mathematical Methods of Operations Research*, 95(1):141–161, 2022.
- [100] D. Gabay and B. Mercier. A dual algorithm for the solution of nonlinear variational problems via finite element approximations. *Computers and Mathematics with Applications*, 2(7):17–40, 1976.
- [101] D. Gale. Neighborly and cyclic polytopes. In *Proceedings of Symposia in Pure Mathematics*, volume 7, pages 225–232, 1963.
- [102] T. Gallai. Über extreme punkt-und kantenmengen. *Annales Universitatis Sci-*

- entiarum Budapestinensis de Rolando Eötvös Nominatae, Sectio Mathematica*, 2:133–138, 1959.
- [103] L. Galli and A. N. Letchford. On the Lovász theta function and some variants. *Discrete Optimization*, 25:159–174, 2017.
- [104] V. Galliard. Classical pseudo telepathy and coloring graphs. Master’s thesis, ETH Zurich, 2001.
- [105] V. Galliard, A. Tapp, and S. Wolf. The impossibility of pseudotelepathy without quantum entanglement. In *IEEE International Symposium on Information Theory, 2003. Proceedings, Yokohama, Japan*, page 457. IEEE, 2003.
- [106] R. Gandhi, M. M. Halldórsson, G. Kortsarz, and H. Shachnai. Improved bounds for sum multicoloring and scheduling dependent jobs with minsum criteria. In *International Workshop on Approximation and Online Algorithms*, pages 68–82. Springer, 2004.
- [107] M. R. Garey and D. S. Johnson. “strong”NP-completeness results: Motivation, examples, and implications. *Journal of the ACM*, 25(3):499–508, 1978.
- [108] K. Gatermann and P. A. Parrilo. Symmetry groups, semidefinite programs, and sums of squares. *Journal of Pure and Applied Algebra*, 192:95–128, 2004.
- [109] P. Gattermann, P. Großmann, K. Nachtigall, and A. Schöbel. Integrating passengers’ routes in periodic timetabling: a SAT approach. In *ATMOS 2016*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2016.
- [110] D. Geller and S. Stahl. The chromatic number and other functions of the lexicographic product. *Journal of Combinatorial Theory, Series B*, 19(1):87–95, 1975.
- [111] A. Ghaffari-Hadigheh, L. Sinjorgo, and R. Sotirov. On convergence of a  $q$ -random coordinate constrained algorithm for non-convex problems. *Journal of Global Optimization*, 90(4):843–868, 2024.
- [112] S. Gharibian and O. Parekh. Almost optimal classical approximation algorithms for a quantum generalization of max-cut. In D. Achlioptas and L. A. Végh, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, volume 145 of *LIPICs*, pages 31:1–31:17, 2019.
- [113] J. C. Gilbert. *SDOlab-A solver of real or complex number semidefinite optimization problems*. PhD thesis, INRIA Paris, 2017.
- [114] J. C. Gilbert and C. Jozs. Plea for a semidefinite optimization solver in complex numbers. Technical report, LAAS–Laboratoire d’analyse et d’architecture des systèmes (Toulouse, France), 2017.
- [115] C. D. Godsil and M. W. Newman. Coloring an orthogonality graph. *SIAM Journal on Discrete Mathematics*, 22(2):683–692, 2008.
- [116] C. D. Godsil and M. W. Newman. Eigenvalue bounds for independent sets. *Journal of Combinatorial Theory, Series B*, 98(4):721–734, 2008.
- [117] M. X. Goemans and D. P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the ACM*, 42(6):1115–1145, 1995.
- [118] M. X. Goemans and D. P. Williamson. Approximation algorithms for MAX-3-CUT and other problems via complex semidefinite programming. *Journal of Computer and System Sciences*, 68(2):442–470, 2004.

- [119] N. I. Gould, M. E. Hribar, and J. Nocedal. On the solution of equality constrained quadratic programming problems arising in optimization. *SIAM Journal on Scientific Computing*, 23(4):1376–1395, 2001.
- [120] J. Gouveia, P. A. Parrilo, and R. R. Thomas. Theta bodies for polynomial ideals. *SIAM Journal on Optimization*, 20(4):2097–2118, 2010.
- [121] N. Graham, H. Hu, J. Im, X. Li, and H. Wolkowicz. A restricted dual Peaceman-Rachford splitting method for a strengthened DNN relaxation for QAP. *INFORMS Journal on Computing*, 2022.
- [122] D. Greenwell and L. Lovász. Applications of product colouring. *Acta Mathematica Academiae Scientiarum Hungaricae*, 25(3-4):335–340, 1974.
- [123] S. Gribling, S. Polak, and L. Slot. A note on the computational complexity of the moment-SOS hierarchy for polynomial optimization. In *Proceedings of the 2023 International Symposium on Symbolic and Algebraic Computation*, pages 280–288, 2023.
- [124] S. Gribling, L. Sinjorgo, and R. Sotirov. Improved approximation ratios for the quantum max-cut problem on general, triangle-free and bipartite graphs. *preprint arXiv:2504.11120*, 2025.
- [125] R. Grone, C. R. Johnson, E. M. Sá, and H. Wolkowicz. Positive definite completions of partial Hermitian matrices. *Linear Algebra and its Applications*, 58: 109–124, 1984.
- [126] R. Grone, S. Pierce, and W. Watkins. Extremal correlation matrices. *Linear Algebra and its Applications*, 134:63–70, 1990.
- [127] M. Grötschel, L. Lovász, and A. Schrijver. The ellipsoid method and its consequences in combinatorial optimization. *Combinatorica*, 1(2):169–197, 1981.
- [128] G. Gruber and F. Rendl. Computational experience with stable set relaxations. *SIAM Journal on Optimization*, 13(4):1014–1028, 2003.
- [129] N. Gvozdenović. *Approximating the stability number and the chromatic number of a graph via semidefinite programming*. PhD thesis, Universiteit van Amsterdam, 2008.
- [130] N. Gvozdenović and M. Laurent. The operator  $\Psi$  for the chromatic number of a graph. *SIAM Journal on Optimization*, 19(2):572–591, 2008.
- [131] W. H. Haemers. An upper bound for the Shannon capacity of a graph. In *Colloq. Math. Soc. János Bolyai*, volume 25, pages 267–272, 1978.
- [132] W. H. Haemers. *Eigenvalue techniques in design and graph theory*. PhD thesis, Technische universiteit Eindhoven, 1979.
- [133] M. M. Halldórsson and G. Kortsarz. Multicoloring: Problems and techniques. In J. Fiala, V. Koubek, and J. Kratochvíl, editors, *Mathematical Foundations of Computer Science 2004*, pages 25–41, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg.
- [134] E. Halperin and U. Zwick. Approximation algorithms for MAX 4-SAT and rounding procedures for semidefinite programs. *Journal of Algorithms*, 40(2): 184–211, 2001.
- [135] J. Håstad. Some optimal inapproximability results. *Journal of the ACM*, 48(4): 798–859, 2001.

- [136] B. He, F. Ma, and X. Yuan. Convergence study on the symmetric version of ADMM with larger step sizes. *SIAM Journal on Imaging Sciences*, 9(3):1467–1501, 2016.
- [137] S. T. Hedetniemi. Homomorphisms of graphs and automata. Technical report, Michigan univ Ann Arbor Communication Sciences Program, 1966.
- [138] D. Henrion and J. Malick. Projection methods for conic feasibility problems: applications to polynomial sum-of-squares decompositions. *Optimization Methods & Software*, 26(1):23–46, 2011.
- [139] D. Henrion, J. B. Lasserre, and J. Löfberg. GloptiPoly 3: moments, optimization and semidefinite programming. *Optimization Methods & Software*, 24(4-5):761–779, 2009.
- [140] D. Hilbert. Über die darstellung definiter formen als summe von formenquadraten. *Mathematische Annalen*, 32(3):342–350, 1888.
- [141] A. Hilton, R. Rado, and S. Scott. A ( $< 5$ )-colour theorem for planar graphs. *Bulletin of the London Mathematical Society*, 5(3):302–306, 1973.
- [142] A. Hoffman. On eigenvalues and colorings of graphs. In B. Harris, editor, *Graph Theory and its Applications*, pages 79–91. Academic Press, New York, 1970.
- [143] R. A. Horn and C. R. Johnson. *Topics in Matrix Analysis*. Cambridge University Press, 1994.
- [144] R. A. Horn and C. R. Johnson. *Matrix Analysis*. Cambridge University Press, Cambridge; New York, 2nd edition, 2013.
- [145] F. Huber, K. Thompson, O. Parekh, and S. Gharibian. Second order cone relaxations for quantum max cut. *preprint arXiv:2411.04120v1*, 2024.
- [146] Y. Hwang, J. Neeman, O. Parekh, K. Thompson, and J. Wright. Unique games hardness of quantum max-cut, and a conjectured vector-valued Borell’s inequality. In *Proceedings of the 2023 Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1319–1384, 2023.
- [147] F. Ihringer and H. Tanaka. The independence number of the orthogonality graph in dimension  $2^k$ . *Combinatorica*, 39(6):1425–1428, 2019.
- [148] I. D. Ivanov and E. de Klerk and. Parallel implementation of a semidefinite programming solver based on CSDP on a distributed memory cluster. *Optimization Methods and Software*, 25(3):405–420, 2010.
- [149] D. Jagt, S. Shivakumar, P. Seiler, and M. Peet. Efficient data structures for exploiting sparsity and structure in representation of polynomial optimization problems: Implementation in SOSTOOLS. *preprint arXiv:2203.01910*, 2022.
- [150] F. Jarre, F. Lieder, Y.-F. Liu, and C. Lu. Set-completely-positive representations and cuts for the max-cut polytope and the unit modulus lifting. *Journal of Global Optimization*, 76(4):913–932, 2020.
- [151] R. Jiang, Y.-F. Liu, C. Bao, and B. Jiang. Tightness and equivalence of semidefinite relaxations for MIMO detection. *preprint arXiv:2102.04586*, 2021.
- [152] M. Jones and J. Anderson. Approximate projections onto the positive semidefinite cone using randomization. *preprint arXiv:2410.19208*, 2024.
- [153] Z. Jorquera, A. Kolla, S. Kordonow, J. S. Sandhu, and S. Wayland. Monogamy of entanglement bounds and improved approximation algorithms for qudit

- Hamiltonians. *preprint arXiv:2410.15544v2*, 2024.
- [154] C. Jozs and D. K. Molzahn. Lasserre hierarchy for large scale polynomial optimization in real and complex variables. *SIAM Journal on Optimization*, 28(2): 1017–1048, 2018.
- [155] N. Ju and A. Nagda. Improved approximation algorithms for the EPR Hamiltonian. *preprint arXiv:2504.10712v1*, 2025.
- [156] I. Kannan, R. King, and L. Zhou. A quantum approximate optimization algorithm for local Hamiltonian problems. *preprint arXiv:2412.09221v1*, 2024.
- [157] H. Karloff and U. Zwick. A 7/8-approximation algorithm for MAX 3SAT? In *Proceedings 38th Annual Symposium on Foundations of Computer Science*, pages 406–415. IEEE, 1997.
- [158] R. M. Karp. Reducibility Among Combinatorial Problems. In R. E. Miller and J. W. Thatcher, editors, *Complexity of Computer Computations*, pages 85–103. Plenum Press, 1972.
- [159] W. Karush. Minima of functions of several variables with inequalities as side constraints. Master’s thesis, University of Chicago, 1939.
- [160] H. Kautz and B. Selman. Unifying SAT-based and graph-based planning. In *IJCAI*, volume 99, pages 318–325, 1999.
- [161] J. Kempe, A. Kitaev, and O. Regev. The complexity of the local Hamiltonian problem. *SIAM Journal on Computing*, 35(5):1070–1097, 2006.
- [162] S. Khot. On the power of unique 2-prover 1-round games. In *Proceedings of the Thirty-Fourth Annual ACM Symposium on Theory of Computing*, STOC ’02, pages 767–775, 2002.
- [163] S. Khot, G. Kindler, E. Mossel, and R. O’Donnell. Optimal inapproximability results for MAX-CUT and other 2-variable CSPs? *SIAM Journal on Computing*, 37(1):319–357, 2007.
- [164] R. King. An improved approximation algorithm for quantum max-cut on triangle-free graphs. *Quantum*, 7:1180, 2023.
- [165] S. Klavžar. Coloring graph products—a survey. *Discrete Mathematics*, 155(1-3):135–145, 1996.
- [166] I. Klep, V. Magron, and J. Povh. Sparse noncommutative polynomial optimization. *Mathematical Programming*, 193(2):789–829, 2022.
- [167] D. E. Knuth. The sandwich theorem. *The Electronic Journal of Combinatorics*, 1(1):A1, 1994.
- [168] A. V. Knyazev. Toward the optimal preconditioned eigensolver: Locally optimal block preconditioned conjugate gradient method. *SIAM journal on scientific computing*, 23(2):517–541, 2001.
- [169] A. Koster and M. Scheffel. A routing and network dimensioning strategy to reduce wavelength continuity conflicts in all-optical networks. In *Proceedings of INOC 2007, International Network Optimization Conference, Spa, Belgium, April 22-25*, 2007.
- [170] A. Kuegel. Improved exact solver for the weighted MAX-SAT problem. *EPiC Series in Computing*, 8:15–27, 2012.
- [171] H. W. Kuhn and A. W. Tucker. Nonlinear programming. In *Berkeley Symposium*

- on Mathematical Statistics and Probability*, pages 481–492. Springer, 1951.
- [172] O. Kuryatnikova, R. Sotirov, and J. C. Vera. The maximum  $k$ -colorable subgraph problem and related problems. *INFORMS Journal on Computing*, 34(1): 656–669, 2022.
- [173] A. Lapidoth. *A foundation in digital communication*. Cambridge University Press, second edition, 2017.
- [174] J. B. Lasserre. An explicit exact SDP relaxation for nonlinear 0-1 programs. In *International Conference on Integer Programming and Combinatorial Optimization*, pages 293–303. Springer, 2001.
- [175] J. B. Lasserre. Global optimization with polynomials and the problem of moments. *SIAM Journal on Optimization*, 11(3):796–817, 2001.
- [176] J. B. Lasserre. A sum of squares approximation of nonnegative polynomials. *SIAM review*, 49(4):651–669, 2007.
- [177] J. B. Lasserre. Convexity in semialgebraic geometry and polynomial optimization. *SIAM Journal on Optimization*, 19(4):1995–2014, 2009.
- [178] J. B. Lasserre. *Moments, positive polynomials and their applications*, volume 1. Imperial College Press, 2009.
- [179] J. B. Lasserre. The moment-SOS hierarchy: Applications and related topics. *Acta Numerica*, 33:841–908, 2024.
- [180] M. Laurent. Graphic vertices of the metric polytope. *Discrete Mathematics*, 151(1-3):131–153, 1996.
- [181] M. Laurent. Tighter linear and semidefinite relaxations for max-cut based on the Lovász–Schrijver lift-and-project procedure. *SIAM Journal on Optimization*, 12(2):345–375, 2002.
- [182] M. Laurent. A comparison of the Sherali-Adams, Lovász-Schrijver, and Lasserre relaxations for 0–1 programming. *Mathematics of Operations Research*, 28(3): 470–496, 2003.
- [183] M. Laurent. Semidefinite relaxations for max-cut. In M. Grötschel, editor, *The sharpest cut: The Impact of Manfred Padberg and his work*, pages 257–290. SIAM, 2004.
- [184] M. Laurent. Semidefinite representations for finite varieties. *Mathematical programming*, 109(1):1–26, 2007.
- [185] M. Laurent. Sums of squares, moment matrices and optimization over polynomials. In *Emerging applications of algebraic geometry*, pages 157–270. Springer, 2009.
- [186] M. Laurent and S. Poljak. On a positive semidefinite relaxation of the cut polytope. *Linear Algebra and its Applications*, 223–224:439–461, 1995.
- [187] M. Laurent and S. Poljak. Gap inequalities for the cut polytope. *European Journal of Combinatorics*, 17(2):233–254, 1996.
- [188] M. Laurent and F. Rendl. *Semidefinite Programming and Integer Programming*, volume 12, chapter 8, pages 393–514. Elsevier, 2005.
- [189] M. Laurent and A. Varvitsiotis. Positive semidefinite matrix completion, universal rigidity and the strong Arnold property. *Linear Algebra and its Applications*, 452:292–317, 2014.

- [190] E. Lee. Optimizing quantum circuit parameters via SDP. In S. W. Bae and H. Park, editors, *33rd International Symposium on Algorithms and Computation, Seoul, Korea*, volume 248 of *LIPICs*, pages 48:1–48:16, 2022.
- [191] E. Lee and O. Parekh. An improved quantum max cut approximation via maximum matching. In K. Bringmann, M. Grohe, G. Puppis, and O. Svensson, editors, *51st International Colloquium on Automata, Languages, and Programming*, volume 297 of *LIPICs*, pages 105:1–105:11, 2024.
- [192] M. Lewin, D. Livnat, and U. Zwick. Improved rounding techniques for the MAX 2-SAT and MAX DI-CUT problems. In *International Conference on Integer Programming and Combinatorial Optimization*, pages 67–82. Springer, 2002.
- [193] J. Lewis and M. Yannakakis. The node-deletion problem for hereditary properties is NP-complete. *Journal of Computer and System Sciences*, 20:219–230, 1980.
- [194] C.-K. Li and B.-S. Tam. A note on extreme correlation matrices. *SIAM Journal on Matrix Analysis and Applications*, 15(3):903–908, 1994.
- [195] C. M. Li and F. Manya. MaxSAT, hard and soft constraints. In A. Biere, M. J. H. Heule, H. van Maaren, and T. Walsh, editors, *Handbook of satisfiability*, volume 336 of *Frontiers in Artificial Intelligence and Applications*, pages 903–927. IOS Press, 2021.
- [196] X. Li, T. K. Pong, H. Sun, and H. Wolkowicz. A strictly contractive Peaceman-Rachford splitting method for the doubly nonnegative relaxation of the minimum cut problem. *Computational optimization and applications*, 78(3):853–891, 2021.
- [197] W. Lin. Multicoloring and Mycielski construction. *Discrete Mathematics*, 308(16):3565–3573, 2008.
- [198] W. Lin, D. D.-F. Liu, and X. Zhu. Multi-coloring the Mycielskian of graphs. *Journal of Graph Theory*, 63(4):311–323, 2010.
- [199] W. Linz.  $l$ -systems and the Lovász number. *Combinatorica*, 45(2):1–24, 2025.
- [200] R. Lippert, R. Schwartz, G. Lancia, and S. Istrail. Algorithmic strategies for the single nucleotide polymorphism haplotype assembly problem. *Briefings in Bioinformatics*, 3(1):23–31, 2002.
- [201] R. Loewy. Extreme points of a convex subset of the cone of positive semidefinite matrices. *Mathematische Annalen*, 253:227–232, 1980.
- [202] J. Löfberg. YALMIP : A toolbox for modeling and optimization in MATLAB. In *In Proceedings of the CACSD Conference*, Taipei, Taiwan, 2004.
- [203] J. Löfberg. Pre-and post-processing sum-of-squares programs in practice. *IEEE transactions on automatic control*, 54(5):1007–1011, 2009.
- [204] L. Lovász. On the Shannon capacity of a graph. *IEEE Transactions on Information theory*, 25(1):1–7, 1979.
- [205] L. Lovász and A. Schrijver. Cones of matrices and set-functions and 0–1 optimization. *SIAM Journal on Optimization*, 1(2):166–190, 1991.
- [206] L. Lovász. *An algorithmic theory of numbers, graphs, and convexity*. SIAM, Philadelphia, PA, 1986.

- [207] C. Lu, Z. Deng, W.-Q. Zhang, and S.-C. Fang. Argument division based branch-and-bound algorithm for unit-modulus constrained complex quadratic programming. *Journal of Global Optimization*, 70:171–187, 2018.
- [208] C. Lu, Y.-F. Liu, W.-Q. Zhang, and S. Zhang. Tightness of a new and enhanced semidefinite relaxation for MIMO detection. *SIAM Journal on Optimization*, 29(1):719–742, 2019.
- [209] C. Lu, Y.-F. Liu, and J. Zhou. An enhanced SDR based global algorithm for nonconvex complex quadratic programs with signal processing applications. *IEEE Open Journal of Signal Processing*, 1:120–134, 2020.
- [210] C. Luo, S. Cai, W. Wu, Z. Jie, and K. Su. CCLS: An efficient local search algorithm for weighted maximum satisfiability. *IEEE Transactions on Computers*, 64(7):1830–1843, 2015.
- [211] R. Madani, A. Kalbat, and J. Lavaei. ADMM for sparse semidefinite programming with applications to optimal power flow problem. In *2015 54th IEEE Conference on Decision and Control*, pages 5932–5939. IEEE, 2015.
- [212] V. Magron and J. Wang. *Sparse polynomial optimization: theory and practice*. World Scientific, 2023.
- [213] V. Magron, M. Safey El Din, M. Schweighofer, and T. H. Vu. Exact SOHS decompositions of trigonometric univariate polynomials with Gaussian coefficients. In *Proceedings of the 2022 International Symposium on Symbolic and Algebraic Computation*, pages 325–332, 2022.
- [214] E. Malaguti and P. Toth. An evolutionary approach for bandwidth multicoloring problems. *European Journal of Operational Research*, 189(3):638–651, 2008.
- [215] M. Marek-Sadowska. An unconstrained topological via minimization problem for two-layer routing. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 3(3):184–190, 1984.
- [216] J. P. Marques-Silva and K. A. Sakallah. Boolean satisfiability in electronic design automation. In *Proceedings of the 37th Annual Design Automation Conference*, pages 675–680, 2000.
- [217] M. Marshall. Optimization of polynomial functions. *Canadian Mathematical Bulletin*, 46(4):575–587, 2003.
- [218] K. Marwaha, A. She, and J. Sud. Performance of variational algorithms for local Hamiltonian problems on random regular graphs. *preprint arXiv:2412.15147v1*, 2024.
- [219] D. Marx. The complexity of tree multicolorings. In R. W. Diks K., editor, *Mathematical Foundations of Computer Science, MFCS*, volume 2420, pages 532–542. Lecture Notes in Computer Science, Springer, Berlin, Heidelberg, 2002.
- [220] S. Matuura and T. Matsui. New approximation algorithms for MAX 2SAT and MAX DICUT. *Journal of the Operations Research Society of Japan*, 46(2):178–188, 2003.
- [221] R. J. McEliece, E. R. Rodemich, and H. C. Rumsey Jr. The Lovász bound and some generalizations. *J. Combin. Inform. System Sci*, 3(3):134–152, 1978.
- [222] B. D. McKay and A. Piperno. Practical graph isomorphism, II. *Journal of Symbolic Computation*, 60:94–112, 2014.
- [223] P. McMullen. The maximum numbers of faces of a convex polytope. *Mathe-*

- matika*, 17(2):179–184, 1970.
- [224] A. Mehrotra and M. A. Trick. A branch-and-price approach for graph multicoloring. In E. K. Baker, A. Joseph, A. Mehrotra, and M. A. Trick, editors, *Extending the Horizons: Advances in Computing, Optimization, and Decision Technologies*, pages 15–29. Springer, Boston, MA, 2007.
- [225] M. Mendonca, A. Wąsowski, and K. Czarnecki. SAT-based analysis of feature models is easy. In *Proceedings of the 13th International Software Product Line Conference*, pages 231–240, 2009.
- [226] A. Mobasher, M. Taherzadeh, R. Sotirov, and A. K. Khandani. A near-maximum-likelihood decoding algorithm for MIMO systems based on semi-definite programming. *IEEE Transactions on Information Theory*, 53(11):3869–3886, 2007.
- [227] R. D. C. Monteiro and P. Zanjácomo. Implementation of primal-dual methods for semidefinite programming based on Monteiro and Tsuchiya Newton directions and their variants. *Optimization Methods and Software*, 11(1-4):91–140, 1999.
- [228] MOSEK ApS. *The MOSEK optimization toolbox for MATLAB manual. Version 11.0*, 2025. URL <http://docs.mosek.com/11.0/toolbox/index.html>.
- [229] E. Mossel, R. O’Donnell, and K. Oleszkiewicz. Noise stability of functions with low influences: invariance and optimality. In *46th Annual IEEE Symposium on Foundations of Computer Science (FOCS’05)*, pages 21–30. IEEE, 2005.
- [230] G. Myerson. How small can a sum of roots of unity be? *The American Mathematical Monthly*, 93(6):457–459, 1986.
- [231] G. Narasimhan. *The maximum  $k$ -colorable subgraph problem*. PhD thesis, University of Wisconsin-Madison, 1989.
- [232] G. Narasimhan and R. Manber. A generalization of Lovász sandwich theorem. Technical report, University of Wisconsin-Madison Department of Computer Sciences, 1988.
- [233] L. Narayanan. Channel assignment and graph multicoloring. In I. Stojmenović, editor, *Handbook of Wireless Networks and Mobile Computing*, chapter 4, pages 71–94. Wiley Online Library, 2002.
- [234] M. Navascués, S. Pironio, and A. Acín. A convergent hierarchy of semidefinite programs characterizing the set of quantum correlations. *New Journal of Physics*, 10, 2008.
- [235] Y. Nesterov. Squared functional systems and optimization problems. In H. Frenk, K. Roos, T. Terlaky, and S. Zhang, editors, *High Performance Optimization*, pages 405–440. Springer, 2000.
- [236] Y. Nesterov and A. Nemirovskii. Interior point polynomial algorithms in convex programming. In *SIAM Studies in Applied Mathematics*. SIAM, Philadelphia, USA, Vol. 13, 1994.
- [237] M. W. Newman. *Independent sets and eigenspaces*. PhD thesis, University of Waterloo, 2004.
- [238] J. Nie. Polynomial optimization with real varieties. *SIAM Journal on Optimization*, 23(3):1634–1646, 2013.
- [239] J. Nie. Optimality conditions and finite convergence of Lasserre’s hierarchy.

- Mathematical programming*, 146:97–121, 2014.
- [240] J. Nie. *Moment and Polynomial Optimization*. MOS-SIAM series on optimization. SIAM, 2023.
- [241] E. A. Nordhaus and J. W. Gaddum. On complementary graphs. *The American Mathematical Monthly*, 63(3):175–177, 1956.
- [242] R. O’Donnell. SOS Is Not Obviously Automatizable, Even Approximately. In C. H. Papadimitriou, editor, *8th Innovations in Theoretical Computer Science Conference (ITCS 2017)*, volume 67 of *LIPICs*, pages 59:1–59:10, 2017.
- [243] D. E. Oliveira, H. Wolkowicz, and Y. Xu. ADMM for the SDP relaxation of the QAP. *Mathematical Programming Computation*, 10(4):631–658, 2018.
- [244] M. L. Overton and R. S. Womersley. Optimality conditions and duality theory for minimizing sums of the largest eigenvalues of symmetric matrices. *Mathematical Programming*, 62(1-3):321–357, 1993.
- [245] A. Papachristodoulou, J. Anderson, G. Valmorbida, S. Prajna, P. Seiler, P. A. Parrilo, M. M. Peet, and D. Jagt. *SOSTOOLS: Sum of squares optimization toolbox for MATLAB*. <http://arXiv.org/abs/1310.4716>, 2021. URL <https://github.com/oxfordcontrol/SOSTOOLS>.
- [246] O. Parekh and K. Thompson. Application of the level-2 quantum Lasserre hierarchy in quantum approximation algorithms. In N. Bansal, E. Merelli, and J. Worrell, editors, *48th International Colloquium on Automata, Languages, and Programming*, volume 198 of *LIPICs*, pages 102:1–102:20, 2021.
- [247] O. Parekh and K. Thompson. An optimal product-state approximation for 2-local quantum Hamiltonians with positive terms. *preprint arXiv:2206.08342*, 2022.
- [248] P. A. Parrilo. An explicit construction of distinguished representations of polynomials nonnegative over finite sets. *Preprint, ETH, Zürich*, 2002.
- [249] P. A. Parrilo. Semidefinite programming relaxations for semialgebraic problems. *Mathematical programming*, 96:293–320, 2003.
- [250] P. A. Parrilo and R. R. Thomas, editors. *Sum of Squares: Theory and Applications*, volume 77 of *Proceedings of Symposia in Applied Mathematics*. American Mathematical Society, Providence, Rhode Island, 2020.
- [251] D. W. Peaceman and H. H. Rachford, Jr. The numerical solution of parabolic and elliptic differential equations. *Journal of the Society for industrial and Applied Mathematics*, 3(1):28–41, 1955.
- [252] F. Permenter and P. A. Parrilo. Basis selection for SOS programs via facial reduction and polyhedral approximations. In *53rd IEEE Conference on Decision and Control*, pages 6615–6620. IEEE, 2014.
- [253] R. M. N. Pesce and P. D. Stevenson. H2ZIXY: Pauli spin matrix decomposition of real symmetric matrices. *preprint arXiv:2111.00627*, 2021.
- [254] H. Peyrl and P. A. Parrilo. Computing sum of squares decompositions with rational coefficients. *Theoretical Computer Science*, 409(2):269–281, 2008.
- [255] S. Piddock and A. Montanaro. The complexity of antiferromagnetic interactions and 2D lattices. *Quantum Information & Computation*, 17(7-8):636–672, 2017.
- [256] S. Pironio, M. Navascués, and A. Acín. Convergent relaxations of polynomial

- optimization problems with noncommuting variables. *SIAM Journal on Optimization*, 20(5):2157–2180, 2010.
- [257] J. Plesník. Finding the orthogonal projection of a point onto an affine subspace. *Linear algebra and its applications*, 422(2-3):455–470, 2007.
- [258] S. Poljak and F. Rendl. Nonpolyhedral relaxations of graph-bisection problems. *SIAM Journal on Optimization*, 5(3):467–487, 1995.
- [259] M. R. Prasad, A. Biere, and A. Gupta. A survey of recent advances in SAT-based formal verification. *International Journal on Software Tools for Technology Transfer*, 7(2):156–173, 2005.
- [260] D. Pucher and F. Rendl. Practical experience with stable set and coloring relaxations. *preprint arXiv:2401.17069v2*, 2024.
- [261] M. Putinar. Positive polynomials on compact semi-algebraic sets. *Indiana University Mathematics Journal*, 42(3):969–984, 1993.
- [262] P. Raghavendra and B. Weitz. On the bit complexity of sum-of-squares proofs. In I. Chatzigiannakis, P. Indyk, F. Kuhn, and A. Muscholl, editors, *44th International Colloquium on Automata, Languages, and Programming*, volume 80 of *LIPICs*, pages 80:1–80:13. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2017.
- [263] G. Ren and Y. Bu.  $k$ -Fold coloring of planar graphs. *Science China Mathematics*, 53(10):2791–2800, 2010.
- [264] B. Reznick. Extremal PSD forms with few terms. *Duke mathematical journal*, 45(2):363–374, 1978.
- [265] R. T. Rockafellar. *Convex analysis*, volume 11. Princeton University Press, 1997.
- [266] N. Rontsis, P. Goulart, and Y. Nakatsukasa. Efficient semidefinite programming with approximate ADMM. *Journal of Optimization Theory and Applications*, 192(1):292–320, 2022.
- [267] G. Sabidussi. Graphs with given group and given graph-theoretical properties. *Canadian Journal of Mathematics*, 9:515–525, 1957.
- [268] R. Sarkar and E. van den Berg. On sets of maximally commuting and anti-commuting Pauli operators. *Research in the Mathematical Sciences*, 8(1):14, 2021.
- [269] H. Sayama. Estimation of Laplacian spectra of direct and strong product graphs. *Discrete Applied Mathematics*, 205:160–170, 2016.
- [270] C. Scheiderer. Positivity and sums of squares: a guide to recent results. In *Emerging applications of algebraic geometry*, pages 271–324. Springer, 2009.
- [271] A. Schrijver. A comparison of the Delsarte and Lovász bounds. *IEEE Transactions on Information Theory*, 25(4):425–429, 1979.
- [272] Z. Shan and E. T. Wang. The gaps between consecutive binomial coefficients. *Mathematics Magazine*, 63(2):122–124, 1990.
- [273] C. Shannon. The zero error capacity of a noisy channel. *IRE Transactions on Information Theory*, 2:8–19, 1956.
- [274] H. D. Sherali and W. P. Adams. A hierarchy of relaxations between the continuous and convex hull representations for zero-one programming problems. *SIAM*

- Journal on Discrete Mathematics*, 3(3):411–430, 1990.
- [275] Y. Shitov. Counterexamples to Hedetniemi’s conjecture. *Annals of Mathematics*, 190(2):663–667, 2019.
- [276] N. Z. Shor. An approach to obtaining global extremums in polynomial mathematical programming problems. *Cybernetics*, 23(5):695–700, 1987.
- [277] F. Silvestri. Spectrahedral relaxations of the stable set polytope using induced subgraphs. Master’s thesis, Universität Heidelberg, 2013.
- [278] A. Singer. Angular synchronization by eigenvectors and semidefinite programming. *Applied and computational harmonic analysis*, 30(1):20–36, 2011.
- [279] R. Singleton. Maximum distance  $q$ -nary codes. *IEEE Transactions on Information Theory*, 10(2):116–118, 1964.
- [280] M. Sinha. Lower bounds for approximating the matching polytope. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1585–1604, 2018.
- [281] L. Sinjorgo and R. Sotirov. On the generalized  $\vartheta$ -number and related problems for highly symmetric graphs. *SIAM Journal on Optimization*, 32(2):1344–1378, 2022.
- [282] L. Sinjorgo and R. Sotirov. On solving MAX-SAT using sum of squares. *INFORMS Journal on Computing*, 36(2):417–433, 2024.
- [283] L. Sinjorgo, R. Sotirov, and M. F. Anjos. Cuts and semidefinite liftings for the complex cut polytope. *Mathematical Programming*, pages 1–50, 2024.
- [284] L. Sinjorgo, R. Sotirov, and J. C. Vera. SDP bounds on the stability number via ADMM and intermediate levels of the Lasserre hierarchy. *preprint arXiv:2506.08648*, 2025.
- [285] M. Slater. Lagrange multipliers revisited. Cowles Commission Discussion Paper, No. 403, 1950.
- [286] A. M.-C. So, J. Zhang, and Y. Ye. On approximating complex quadratic optimization problems via semidefinite programming relaxations. *Mathematical Programming*, 110(1):93–110, 2007.
- [287] M. Soltanalian and P. Stoica. Designing unimodular codes via quadratic optimization. *IEEE Transactions on Signal Processing*, 62:1221–1234, 2013.
- [288] S. Stahl.  $n$ -Tuple colorings and associated graphs. *Journal of Combinatorial Theory, Series B*, 20(2):185–203, 1976.
- [289] J. F. Sturm. Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. *Optimization methods and software*, 11(1-4):625–653, 1999.
- [290] B. Sturmfels. Polynomial equations and convex polytopes. *The American mathematical monthly*, 105(10):907–922, 1998.
- [291] A. P. Subramanian, H. Gupta, S. R. Das, and M. M. Buddhikot. Fast spectrum allocation in coordinated dynamic spectrum access based cellular networks. In *2007 2nd IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks*, pages 320–330, 2007.
- [292] S. Szabó and B. Zaválnij. Benchmark problems for exhaustive exact maximum clique search algorithms. *Informatika*, 43(2):177–186, 2019.
- [293] J. Takahashi, C. Rayudu, C. Zhou, R. King, K. Thompson, and O. Parekh. An

- SU(2)-symmetric semidefinite programming hierarchy for quantum max cut. *preprint arXiv:2307.15688v2*, 2023.
- [294] G. J. Tee. Eigenvectors of block circulant and alternating circulant matrices. *New Zealand Journal of Mathematics*, 36(8):195–211, 2007.
- [295] The Sage Developers. *SageMath, the Sage Mathematics Software System (Version 10.3)*, 2024. <https://www.sagemath.org>.
- [296] M. J. Todd. Semidefinite optimization. *Acta Numerica*, 10:515–560, 2001.
- [297] M. J. Todd. On max- $k$ -sums. *Mathematical Programming*, 171(1):489–517, 2018.
- [298] H. van Maaren and L. van Norden. Sums of squares, satisfiability and maximum satisfiability. In *International Conference on Theory and Applications of Satisfiability Testing*, pages 294–308. Springer, 2005.
- [299] H. van Maaren, L. van Norden, and M. J. H. Heule. Sums of squares based approximation algorithms for MAX-SAT. *Discrete Applied Mathematics*, 156(10):1754–1779, 2008.
- [300] L. Vandenberghe and S. Boyd. Semidefinite programming. *SIAM review*, 38(1):49–95, 1996.
- [301] V. G. Vizing. On an estimate of the chromatic class of a  $p$ -graph. *Discret. Analiz.*, 3:25–30, 1964.
- [302] H. Waki, S. Kim, M. Kojima, and M. Muramatsu. Sums of squares and semidefinite program relaxations for polynomial optimization problems with structured sparsity. *SIAM Journal on Optimization*, 17(1):218–242, 2006.
- [303] I. Waldspurger, A. d’Aspremont, and S. Mallat. Phase recovery, MaxCut and complex semidefinite programming. *Mathematical Programming*, 149(1-2):47–81, 2015.
- [304] J. Wang. A more efficient reformulation of complex SDP as real SDP. *preprint arXiv:2307.11599v3*, 2024.
- [305] J. Wang and V. Magron. Exploiting sparsity in complex polynomial optimization. *Journal of Optimization Theory and Applications*, pages 1–25, 2022.
- [306] J. Wang and V. Magron. A real moment-HSOS hierarchy for complex polynomial optimization with real coefficients. *Computational Optimization and Applications*, 90(1):53–75, 2025.
- [307] J. Wang, V. Magron, and J. B. Lasserre. Chordal-TSSOS: a moment-SOS hierarchy that exploits term sparsity with chordal extension. *SIAM Journal on Optimization*, 31(1):114, 2021.
- [308] J. Wang, V. Magron, and J. B. Lasserre. TSSOS: A moment-SOS hierarchy that exploits term sparsity. *SIAM Journal on Optimization*, 31(1):30–58, 2021.
- [309] J. Wang, V. Magron, and N. H. A. Mai. CS-TSSOS: Correlative and term sparsity for large-scale polynomial optimization. *ACM Transactions on Mathematical Software*, 48(4):1–26, 2022.
- [310] P.-W. Wang and J. Zico Kolter. Low-rank semidefinite programming for the MAX2SAT problem. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 1641–1649, 2019.
- [311] A. B. Watts, A. Chowdhury, A. Epperly, J. W. Helton, and I. Klep. Relaxations

- and exact solutions to quantum max cut via the algebraic structure of swap operators. *Quantum*, 8:1352, 2024.
- [312] A. Wiegele and S. Zhao. SDP-based bounds for graph partition via extended ADMM. *Computational Optimization and Applications*, 82(1):251–291, 2022.
- [313] H. Wolkowicz, R. Saigal, and L. Vandenberghe, editors. *Handbook of semidefinite programming: theory, algorithms, and applications*. Springer Science & Business Media, 2012.
- [314] S. Yang and L. Hanzo. Fifty years of MIMO detection: The road to large-scale MIMOs. *IEEE Communications Surveys & Tutorials*, 17:1941–1988, 2015.
- [315] M. Yannakakis and F. Gavril. The maximum  $k$ -colorable subgraph problem for chordal graphs. *Information Processing Letters*, 24(2):133 – 137, 1987.
- [316] S. Zhang and Y. Huang. Complex quadratic optimization and semidefinite programming. *SIAM Journal on Optimization*, 16(3):871–890, 2006.
- [317] Y. J. A. Zhang and A. M.-C. So. Optimal spectrum sharing in MIMO cognitive radio networks via semidefinite programming. *IEEE Journal on Selected Areas in Communications*, 29(2):362–373, 2011.
- [318] Y. Zheng, G. Fantuzzi, and A. Papachristodoulou. Exploiting sparsity in the coefficient matching conditions in sum-of-squares programming using ADMM. *IEEE control systems letters*, 1(1):80–85, 2017.
- [319] Y. Zhong and N. Boumal. Near-optimal bounds for phase synchronization. *SIAM Journal on Optimization*, 28(2):989–1016, 2018.

## CentER Dissertation Series

CentER for Economic Research, Tilburg University, the Netherlands

No.	Author	Title	ISBN	Published
756	Wanqing Zhang	Influence of Stress, Perceived Control, and Intrinsic Motivation on Individual Economic Decision-Making	978 90 5668 758 8	January 2025
757	Tijn Fleuren	Stochastic Approaches for Production-Inventory Planning: Applications to High-Tech Supply Chains	978 90 5668 759 5	January 2025
758	Wim Maas	Balancing Acts: Executive Compensation, Governance, and Accountability in Nonprofit Organizations	978 90 5668 760 1	January 2025
759	Susanne van der Velden	WAVELENGTHS: Unravelling ASML's sources of innovation	978 90 5668 761 8	February 2025
760	Cardin Masselink	A Manager's Perspective on Feedback and Control Decisions	978 90 5668 762 5	February 2025
761	Lieke Beekers	Essays on Socioeconomic Inequalities	978 90 5668 763 2	April 2025
762	Wei Yao	The US Quantitative Easing Monetary Policy and Commodities' Prices	978 90 5668 764 9	April 2025
763	Jun-Hee An	Essays on Pension Economics and Individual Welfare	978 90 5668 765 6	April 2025
764	Steffen Wolfer	Innovation and corporate catching-up in China: A configurational approach	978 90 5668 766 3	May 2025
765	Elisa Castagno	Essays on Household Financial Decision-Making	978 90 5668 767 0	May 2025
766	Quang Phúc Phùng	Three Essays in Experimental Economics	978 90 5668 768 7	May 2025
767	Ruonan Fu	Essays on Ambiguity, Market Incompleteness, and Asset Pricing	978 90 5668 769 4	May 2025
768	Giuseppe Floccari	Essays in Household Finance and Macroeconomics	978 90 5668 770 0	June 2025

<b>No.</b>	<b>Author</b>	<b>Title</b>	<b>ISBN</b>	<b>Published</b>
769	Hulai Zhang	Essays on the Real Effects of Financial Markets and Sustainable Investments	978 90 5668 776 2	June 2025
770	Fatma Sueda Evirgen	The Economics of Vulnerability and Discrimination	978 90 5668 772 4	June 2025
771	Gizem Taş	Towards Enhanced Deep Learning for Epistasis Modeling: Applications to Genetics of Amyotrophic Lateral Sclerosis	978 90 5668 773 1	June 2025
772	Jierui Yang	Reciprocity and Coordination	978 90 5668 774 8	June 2025
773	Joep van der Plas	Channel Blurring: National Brands at Hard Discounters	978 90 5668 775 5	June 2025
774	Jurian Hendrikse	Essays on Non-Financial Transparency and Performance	978 90 5668 777 9	July 2025
775	Selin Arslanoğlu	Institutions, Communication, and Identity: Experiments in Cooperation, Coordination, and Compliance	978 90 5668 771 7	September 2025
776	Zihao Liu	Empirical Corporate Finance and Deep Learning	978 90 5668 778 6	September 2025
777	Shobhit Kulshreshtha	Essays on Regional Variation in Health and Healthcare Utilization	978 90 5668 779 3	September 2025
778	Christos Revelas	Essays on consistency and randomization in machine learning and fraud detection	978 90 5668 780 9	September 2025
779	Giovanni Trebbi	Natural Language Processing in Finance and Empirical Macroeconomics	978 90 5668 781 6	October 2025
780	Arjan Bruil	From Macro Totals to Household Distributions: Advancing the National Accounts with Inequality Metrics	978 90 5668 782 3	October 2025
781	Kadircan Çakmak	Essays on Consumer Search, Product Returns, and Pricing in Online Markets	978 90 5668 783 0	October 2025
782	Lieske Coumans	Robust Bond Investment Strategies under Parameter Uncertainty	978 90 5668 784 7	November 2025
783	Hyo Eun Sarina Son	Essays on Firms' and Stakeholders' Responses to CSI Incidents	978 90 5668 785 4	December 2025

<b>No.</b>	<b>Author</b>	<b>Title</b>	<b>ISBN</b>	<b>Published</b>
784	Ceren Şahin	Essays on Consumer Anticipation: How Anticipated Feelings and Judgments Shape Consumer Decisions Across Sustainable, Ethical, and Hedonic Consumption	978 90 5668 786 1	December 2025
785	Jan Fredy Agustin Sandoval	Essays on Empirical Asset Pricing	978 90 5668 787 8	December 2025
786	Mario Martini	Approaches to Cooperative Compliance in the BRIC countries and in international programs: Identifying, reviewing and testing elements and approaches to Cooperative Compliance in the BRIC countries and in international programs	978 90 5668 788 5	February 2026
787	Tal Strauss	Cyber Risk, Regulation, and Firm Resilience	978 90 5668 789 2	February 2026
788	Hong Phuoc Michael Vo	Priced to Perfection? Subjective Expectations in Financial Markets	978 90 5668 790 8	February 2026
789	Lennart Sinjorgo	Semidefinite Programming Approaches for Stable Set and Max-Cut Problems	978 90 5668 791 5	February 2026



This thesis investigates the use of semidefinite programming (SDP) for solving (variants of) the well-known stable set and max-cut problems. Chapter 2 considers a generalization of the stable set problem, and a corresponding SDP relaxation. Similarly, Chapter 3 considers a generalization of the max-cut problem that involves complex roots of unity. Various complex SDP relaxations of this generalized max-cut problem are studied. Chapter 4 provides approximation algorithms for the quantum generalization of the max-cut problem. These approximation algorithms employ a hierarchy of SDP relaxations for noncommutative polynomial optimization problems. Chapter 5 provides an SDP algorithm for computing bounds on the stability numbers of graphs. Chapter 6 provides an SDP algorithm for solving the MAX-SAT problem. Chapters 5 and 6 provide extensive numerical results on these algorithms.

**LENNART SINJORGIO ('S-HERTOGENBOSCH, THE NETHERLANDS, 1998)** received his bachelor's, master's, and research master's degrees in operations research from Tilburg University in 2019, 2021, and 2022 respectively. In September 2022, he started as a PhD candidate in operations research at Tilburg University.

ISBN: 978 90 5668 799 15

DOI: 10.26116/tisem.63859127