Centrum voor Wiskunde en Informatica

REPORT RAPPORT

PNA

Probability, Networks and Algorithms

Probability, Networks and Algorithms

Semidefinite programming and integer programming

M. Laurent, F. Rendl

REPORT PNA-R0210 APRIL 30, 2002

P.O. Box 94079, 1090 GB Amsterdam (NL)
Kruislaan 413, 1098 SJ Amsterdam (NL)
Telephone +31 20 592 9333
Telefax +31 20 592 4199

ISSN 1386-3711

Semidefinite Programming and Integer Programming

Monique Laurent * and Franz Rendl **

** CWI, Kruislaan 413, 1098 SJ Amsterdam, The Netherlands*
monique@cwi.nl
*** Universität Klagenfurt, Institut für Mathematik*
Universitätsstrasse 65-67, A-9020 Klagenfurt, Austria
franz.rendl@uni-klu.ac.at

ABSTRACT

We survey how semidefinite programming can be used for finding good approximative solutions to hard combinatorial optimization problems.

2000 Mathematics Subject Classification: 52B12, 68W25, 90C05, 90C22, 90C27.
Keywords and Phrases: combinatorial optimization, linear programming, semidefinite programming, approximative algorithm.
Note: The research of the first author was carried out under the PNA1.1-project of CWI.

Contents

1 Introduction **3**

2 Semidefinite Programming: Duality, Algorithms, Complexity, and Geometry **3**
2.1 Duality . . . 3
2.2 Algorithms . . . 7
2.3 Complexity . . . 9
2.4 Geometry . . . 11

3 Semidefinite Programming Versus Integer 0/1 Programming **15**
3.1 A general paradigm . . . 15
3.2 Introduction on cutting planes and lift-and-project methods . . . 17
3.3 A canonical lifting construction . . . 19
3.4 The Balas-Ceria-Cornuéjols, Lovász-Schrijver, Sherali-Adams, and Lasserre methods . 20
3.5 Application to the stable set problem . . . 23
3.6 Application to the max-cut problem . . . 25
3.7 Extensions to polynomial programming . . . 27

4 Semidefinite Relaxation for the Maximum Stable Set Problem **30**
4.1 The basic linear relaxation . . . 30
4.2 The theta function $\vartheta(G)$ and the basic semidefinite relaxation TH(G) . . . 31
4.3 Sharpening the theta function . . . 34
4.4 Colouring and finding maximum stable sets in perfect graphs . . . 36

5 Semidefinite Relaxation for the Max-Cut Problem 37
5.1 The basic linear relaxation . . . 37
5.2 The basic semidefinite relaxation . . . 38
5.3 The Goemans-Williamson randomized approximation algorithm for max-cut . . . 40
5.4 How to improve the Goemans-Williamson algorithm? . . . 42

6 Applications of Semidefinite Programming and the Rounding Hyperplane Technique to Other Combinatorial Optimization Problems 45
6.1 Approximating quadratic programming . . . 45
6.2 Approximating the maximum bisection problem . . . 47
6.3 Approximating the max k-cut problem . . . 50
6.4 Approximating graph colouring . . . 52
6.5 Approximating the maximum stable set and vertex cover problems . . . 54
6.6 Approximating MAX SAT . . . 56
6.7 Approximating the maximum directed cut problem . . . 62

7 Further Topics 65
7.1 Approximating polynomial programming using semidefinite programming . . . 65
7.2 Approximating combinatorial problems using copositive programming . . . 68

8 Semidefinite Programming and the Quadratic Assignment Problem 70

9 Epilogue: Semidefinite Programming and Algebraic Connectivity 74

1 Introduction

Linear optimization is a relatively young area of applied mathematics. Even though the world is nonlinear, as physicists never stop to point out, it seems that in many practical situations a linearized model describes key features of a problem quite accurately.

The success of linear optimization in many real-world applications has led to the study of integer linear programming, which permits to model optimal decision making under finitely many alternatives. A natural way to approach these types of problems consists in using again linear theory, in this case polyhedral combinatorics, to solve them. Mathematically, one tries to find (at least) a (partial) linear description of the convex hull of all integral solutions. While this approach was succesful for many combinatorial optimization problems, it turned out that some graph optimization problems, such as Max-Cut or Max-Clique, can not be approximated tightly by purely linear methods.

Stronger relaxation methods have therefore attracted the focus of recent research. The extension of linear optimization to semidefinite optimization has turned out to be particularly interesting for the following reasons. First, algorithmic ideas can be extended quite naturally from linear to semidefinite optimization. Secondly, there is theoretical evidence that semidefinite models are sometimes significantly stronger than purely linear ones, justifying the computational overhead to solve them.

It is the purpose of this chapter to explain in detail how semidefinite programming is used to solve integer programming problems. Specifically, we start out in the next section with explaining the relevant mathematical background underlying semidefinite programming by summarizing the necessary duality theory, explaining algorithmic ideas and recalling computational complexity results related to semidefinite programming. In Section 3 we show how semidefinite relaxations arise from integer 0/1 programming by lifting the problem formulated in $\mathbb{R}^n$ to a problem in the space of symmetric matrices.

A detailed study of two prominent special graph optimization problems follows in Sections 4, dealing with the stable set problem, and Section 5, devoted to Max-Cut. For both these problems the extension of polyhedral to semidefinite relaxations has led to a significant improvement in the approximation of the original problem. Section 5 also introduces the hyperplane rounding idea of Goemans and Williamson, which opened the way to many other approximation approaches, many of which are discussed in Section 6.

Section 7 discusses possible alternatives to the use of semidefinite models to get stronger relaxations of integer programs.

Finally, we summarize in Section 8 some recent semidefinite and other nonlinear relaxations applied to the Quadratic Assignment Problem, which have led to a computational breack-through in Branch and Bound computations for this problem.

2 Semidefinite Programming: Duality, Algorithms, Complexity, and Geometry

2.1 Duality

To develop a duality theory for semidefinite programming problems, we take a more general point of view, and look at *Linear Programs over Cones.*

Suppose K is a closed convex cone in $\mathbb{R}^n$, $c \in \mathbb{R}^n, b \in \mathbb{R}^m$ and A is an $m \times n$ matrix. The problem

$$p^* := \sup\{c^T x : Ax = b,\ x \in K\} \tag{1}$$

is called **Cone-LP**, because we optimize a linear function subject to linear equations, and we have the condition that the decision variables x lie in the cone K.

The dual cone K^* is defined as follows:

$$K^* := \{y \in \mathbb{R}^n : y^T x \geq 0 \ \forall x \in K\}.$$

It is a well known fact, not hard to verify, that K^* is also a closed convex cone.

We will derive the dual of (1) by introducing Lagrange multipliers for the equality constraints and using the Minimax Inequality. Let $y \in \mathbb{R}^m$ denote the Lagrange multipliers for $Ax = b$. Using the Lagrangian $L(x, y) := c^T x + y^T(b - Ax)$ we get

$$\inf_y L(x, y) = \begin{cases} c^T x & \text{if } Ax = b \\ -\infty & \text{otherwise} \end{cases}$$

Therefore,

$$p^* = \sup_{x \in K} \inf_y L(x, y) \leq \inf_y \sup_{x \in K} L(x, y).$$

The inequality is usually called 'Minimax inequality', and holds for any real-valued function $L(x, y)$ where x and y are from some ground sets X and Y respectively.

We can rewrite L as $L = b^T y - x^T(A^T y - c)$. The definition of K^* implies the following. If $A^T y - c \notin K^*$ then there exists $x \in K$ such that $x^T(A^T y - c) < 0$. Therefore we conclude

$$\sup_{x \in K} L(x, y) = \begin{cases} b^T y & \text{if } A^T y - c \in K^* \\ \infty & \text{otherwise} \end{cases}$$

This translates into

$$p^* \leq \inf\{b^T y : y \in \mathbb{R}^m, A^T y - c \in K^*\} =: d^*. \tag{2}$$

The problem on the right side of the inequality sign is again a Cone-LP, but this time over the cone K^*. We call this problem the dual to (1). By construction, a pair of dual cone-LP satisfies weak duality.

Lemma 1. (Weak Duality) *Let $x \in K, y \in \mathbb{R}^m$ be given with $Ax = b, A^T y - c \in K^*$. Then, $c^T x \leq b^T y$.*

One crucial issue in duality theory consists in identifying sufficient conditions that insure equality in (2), also called *Strong Duality.* The following condition insures strong duality. We say that the cone-LP (1) satisfies the *Slater constraint qualification* if there exists $x \in int(K)$ such that $Ax = b$. (A similar definition holds for the dual problem.) Duffin [58] shows the following result.

Theorem 2. *If (1) satisfies the Slater constraint qualification and p^* is finite, then $p^* = d^*$, and the dual infimum is attained.*

Returning to semidefinite programs, we consider the vector space $\mathcal{S}_n$ of symmetric $n \times n$ matrices as the ground set for the primal problem. It is equipped with the usual inner product $\langle X, Y \rangle = \mathrm{Tr}(XY)$ for $X, Y \in \mathcal{S}_n$. In this case, the linear operator A, mapping symmetric matrices into $\mathbb{R}^m$ is most conveniently represented by $A(X)_i := \mathrm{Tr}(A_i X)$ for given symmetric matrices $A_i, i = 1, \ldots, m$. The adjoint in this case has the representation $A^T(y) = \sum y_i A_i$. From Fejer's Theorem, which states that

$$A \succeq 0 \text{ if and only if } \mathrm{Tr}(AB) \geq 0 \ \forall B \succeq 0,$$

we see that the cone of positive semidefinite matrices is selfdual. Hence we arrive at the following primal-dual pair of semidefinite programs:

$$\max\{\mathrm{Tr}(CX) : A(X) = b, X \succeq 0\}, \tag{3}$$

$$\min\{b^T y : A^T(y) - C \succeq 0\}. \tag{4}$$

In our combinatorial applications, we usually have the property that both the primal and the dual problem satisfy the Slater constraint qualification, hence we have strong duality and both optima are attained.

Stronger duals for semidefinite programs have been introduced having the property that there is no duality gap, in particular, by Borwein and Wolkowicz [37], Ramana [169] (see [171] for a comparison). In Section 2.3, we will come back briefly to the implications for the complexity of semidefinite programming.

The semidefiniteness of a matrix X can equivalently be expressed as X having only nonnegative eigenvalues. Thus there is some close connection between semidefinite programs and spectral theory of matrices. The following simple examples of semidefinite programs throw some more light onto this connection.

Example 3. *Let C be a symmetric matrix. Consider*

$$\max \mathrm{Tr}(CX) \textit{ such that } \mathrm{Tr}(X) = 1,\ X \succeq 0.$$

The dual is

$$\min y \textit{ such that } yI - C \succeq 0.$$

Both problems clearly satisfy the Slater constraint qualification. In fact, dual feasibility implies that $y \geq \lambda_{\max}(C)$, hence at the optimum $y = \lambda_{\max}(C)$. It is in fact well known that the primal semidefinite program is equivalent to

$$\max x^T C x \textit{ such that } x^T x = 1,$$

by taking $X = xx^T$.

Example 4. *More generally, the sum $\lambda_1 + \ldots + \lambda_k$ of the k largest eigenvalues of $C \in \mathcal{S}_n$ can be expressed as the optimum value of the following semidefinite program:*

$$\max\ \mathrm{Tr}(CX) \quad \textit{such that } I \succeq X \succeq 0,\ \mathrm{Tr}(X) = k \tag{5}$$

which is equivalent to

$$\max\ \mathrm{Tr}(CYY^T) \quad \textit{such that } Y \textit{ is an } n \times k \textit{ matrix with } Y^T Y = I_k. \tag{6}$$

The fact that $\lambda_1 + \ldots + \lambda_k$ is equal to the optimum value of (6) is known as Fan's theorem; see [154] for discussion.

Let us sketch the proof. The fact that the optimum values of the two programs (5) and (6) are equal follows from a nice geometric property of the feasible set of (5) (namely, that its extreme points correspond to the feasible solutions of (6); cf. Lemma 7 below). Let $y_1, \ldots, y_k$ be a set of orthonormal eigenvectors of C for its k largest eigenvalues and let Y be the matrix with columns $y_1, \ldots, y_k$. Then

Y is feasible for (6) and $\mathrm{Tr}(CYY^T) = \sum_{i=1}^k \mathrm{Tr}(y_i^T C y_i) = \sum_{i=1}^k \lambda_i$, which shows that $\sum_{i=1}^k \lambda_i$ is less than or equal to the maximum of (6). Conversely, let Y be an $n \times k$ matrix such that $Y^TY = I_k$; we show that $\mathrm{Tr}(CYY^T) \leq \sum_{i=1}^k \lambda_i$. For this, let $C = Q^T D Q$ where $Q \in \mathcal{S}_n$ with $Q^TQ = I_n$ and $D := \mathrm{diag}(\lambda_1, \ldots, \lambda_n)$. Set $Z := QY$ and $X := ZZ^T$. As Z is an $n \times k$ matrix with $Z^TZ = I_k$, it follows that the only nonzero eigenvalue of X is 1 with multiplicity k and thus X is feasible for (5). Hence, $\mathrm{Tr}(CYY^T) = \mathrm{Tr}(DX) = \sum_{i=1}^n \lambda_i x_{ii} \leq \sum_{i=1}^k \lambda_i$ since $0 \leq x_{ii} \leq 1$ for all i.

By taking the dual of the semidefinite program (5), we obtain the following alternative formulation for the sum of k largest eigenvalues of C:

$$\lambda_1 + \ldots + \lambda_k = \min\ kz + \mathrm{Tr}(Z) \quad \textit{such that } zI + Z \succeq C,\ Z \succeq 0. \tag{7}$$

This latter formulation permits to derive the following semidefinite programming characterization for minimizing the sum of the k largest eigenvalues of a symmetric matrix staifying linear constraints (cf. [5]):

$$\min\ \lambda_1(X) + \ldots + \lambda_k(X) \quad \textit{such that } X \in \mathcal{S}_n,\ \mathrm{Tr}(A_jX) = b_j\ (j = 1, \ldots, m)$$

$$= \min\ kz + \mathrm{Tr}(Z) \quad \textit{such that } zI + Z - X \succeq 0,\ Z \succeq 0,\ \mathrm{Tr}(A_jX) = b_j\ (j = 1, \ldots, m).$$

More recently, Anstreicher and Wolkowicz showed a strong connection between a theorem of Hoffman and Wielandt and semidefinite programming.

Theorem 5. (Hoffman and Wielandt [93]) *Let A and B be symmetric matrices of order n with spectral decomposition $A = PDP^T$, $B = QEQ^T$. We assume that the diagonal matrix D contains the eigenvalues of A in nondecreasing order, and E contains the eigenvalues of B in nonincreasing order. Furthermore, $PP^T = QQ^T = I$. Then*

$$\min\{\mathrm{Tr}(AXBX^T) : X^TX = I\} = \mathrm{Tr}(DE). \tag{8}$$

Moreover, the minimum is attained for $X = PQ^T$.

A proof of this theorem can be found for instance in [93], the result can be traced back to the work of John von Neumann [152]. Anstreicher and Wolkowicz [14] have recently shown that the nonconvex quadratic minimization problem (5) over the set of orthogonal matrices can equivalently be expressed through semidefinite programming. This connection will be a useful tool to bound the Quadratic Assignment Problem, so we recall how this connection can be established. We have:

$$\mathrm{Tr} DE = \min\{\mathrm{Tr} AXBX^T : XX^T = I\} = \min\{\mathrm{Tr} DYEY^T : YY^T = I\}$$

The second equation follows because the mapping $Y = P^TXQ$ is a bijection on the set of orthogonal matrices. We next introduce Lagrange multipliers S and T for the equations $YY^T = I, Y^TY = I$, and get

$$\mathrm{Tr} DE = \min_X \max_{S,T} \mathrm{Tr}(DXEX^T + S(I - XX^T) + T(I - X^TX))$$

$$\geq \max_{S,T} \min_{x=vec(X)} \mathrm{Tr} S + TrT + x^T(E \otimes D - I \otimes S - T \otimes I)x.$$

If $X = (x_1, \ldots, x_n)$ is a matrix with columns x_i, we define $x = vec(X) = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}$ to be the vector obtained from stacking the columns of X. The vec-operator leads to the following identity, see [94].

$$vec(AXB) = (B^T \otimes A)vec(X). \tag{9}$$

$A \otimes B$ denotes the Kronecker product of A and B. Formally,

$$A \otimes B = (a_{ij}B).$$

The inner minimization is bounded only if $E \otimes D - I \otimes S - T \otimes I \succeq 0$. Since D and E are diagonal, we may restrict also S and T to be diagonal, $S = diag(s), T = diag(t)$. This leads to

$$\mathrm{Tr}DE \geq \max\{\sum s_i + \sum t_i : d_i e_j - s_i - t_j \geq 0 \ \forall i, j\}.$$

The last problem is the dual of the assignment problem. Therefore we get

$$\mathrm{Tr}DE \geq \min\{\sum_{ij} d_i e_j z_{ij} : Z = (z_{ij}) \text{ doubly stochastic } \} = \mathrm{Tr}DE.$$

The first term equals the last, so there must be equality throughout. We summarize this as follows.

Theorem 6. [14] *Let A and B be symmetric matrices. Then,*

$$\min\{\mathrm{Tr}AXBX^T : XX^T = I\} = \max\{\mathrm{Tr}S + \mathrm{Tr}T : B \otimes A - I \otimes S - T \otimes I \succeq 0\}.$$

2.2 Algorithms

Semidefinite programs are convex minimization problems, hence they can be solved in polynomial time to any fixed prescribed precision using for instance the ellipsoid method [80]. More recently, interior point methods have turned out to be the method of choice to solve SDP, since they give faster algorithms than the ellipsoid method whose running time is prohibitively high in practice; see for instance the monograph [194].

We will now review the main ideas underlying the interior point approach for SDP. The basic assumption is that both the primal (3) and the dual (4) problem satisfy the Slater constraint qualification, which means we assume that there exists a triple (X, y, Z) such that

$$X \succ 0, \ Z \succ 0, \ A(X) = b, \ Z = A^T(y) - C.$$

To avoid trivialities, it is usually also assumed that the linear equations $A(X) = b$ are linearly independent. In view of Theorem 2, we get the following necessary and sufficient optimality conditions.

A triple (X, y, Z) solves (3) and (4) if and only if

$$A(X) = b, \ X \succeq 0 \qquad \text{(primal feasibility)} \tag{10}$$

$$A^T(y) - Z = C, \ Z \succeq 0 \qquad \text{(dual feasibility)} \tag{11}$$

$$ZX = 0 \qquad \text{(complementarity)} \tag{12}$$

To see how (12) follows from Theorem 2, we note that both the primal and the dual optimum are attained, and the duality gap is 0. If (X, y, Z) is optimal, we get

$$0 = b^T y - \mathrm{Tr} CX = y^T(A(X)) - \mathrm{Tr} CX = \mathrm{Tr}(A^T(y) - C)X = \mathrm{Tr} ZX.$$

Since $X \succeq 0,\ Z \succeq 0$, we also have $X = UU^T,\ Z = VV^T$, for U and V of appropriate size. Thus

$$0 = \mathrm{Tr} ZX = \mathrm{Tr} VV^T UU^T = \|V^T U\|_F^2,$$

hence $V^T U = 0$, so that $ZX = VV^T UU^T = 0$.

In the interior point approach, the condition $ZX = 0$ is replaced by $ZX = \mu I$, leading to a parameterized system of equations:

$$F_\mu(X, y, Z) := \begin{pmatrix} A(X) - b \\ Z - A^T(y) + C \\ ZX - \mu I \end{pmatrix} = 0. \tag{13}$$

Under our assumptions, there exists a unique solution (X, y, Z) for every $\mu > 0$, see for instance [194] (Chapter 10). (To get this result, one interprets (13) as the KKT system of a convex problem with strictly convex cost function.) Denoting this solution by (X_μ, y_μ, Z_μ), it is not too hard to show that the set

$$\{(X_\mu, y_\mu, Z_\mu) : \mu > 0\}$$

defines a smooth curve parameterized by μ, which is usually called the 'central path'.

The interior point approach, more precisely the 'primal-dual interior-point path-following method' consists in applying Newton's method to follow this curve until $\mu \to 0$. This sounds straightforward, and it is, except for the following aspect. The equation (13) has $2\binom{n+1}{2}+m$ variables, but $\binom{n+1}{2}+n^2+m$ equations. The difference arises from $ZX - \mu I$, which need not be symmetric, even if X and Z are. Therefore, some sort of symmetrization of the last equation in (13) is necessary to overcome this problem.

The first papers exploiting this approach [90, 112] use some ad-hoc ideas to symmetrize the last equation. Later, Monteiro [145] and Zhang [199] introduced a rather general scheme to deal with the equation $ZX = \mu I$. Let P be invertible. Zhang considers the mapping $H_P(M) := \frac{1}{2}\left[PMP^{-1} + (PMP^{-1})^T\right]$ and shows that for $X \succ 0,\ Z \succ 0$

$$H_P(ZX) = \mu I \text{ if and only if } ZX = \mu I.$$

Of course, different choices for P produce different search directions after replacing $ZX = \mu I$ by $H_P(ZX)) = \mu I$. Various choices for P have been proposed and investigated with respect to theoretical properties and behaviour in practise.

Todd [189] reviews about 20 different variants for the choice of P and investigates some basic theoretical properties of the resulting search directions. The main message seems to be at present that there is no clear champion among these choices in the sense that it would dominate both with respect to theoretical convergence properties and practical efficiency.

The following variant was introduced by [90], and independently by [112]. It is simple, and yet computationally quite efficient. To simplify the presentation, we assume that there is some starting triple (X, y, Z) which satisfies $A(X) = b,\ A^T(y) - Z = C$ and $X \succ 0,\ Z \succ 0$. If this triple would lie on the central path, its 'path parameter' μ would be $\mu = \frac{1}{n}\mathrm{Tr} ZX$. We do not assume that it lies on the central path, but would like to move from this triple towards the central path, and follow it until $\mu \approx 0$. Therefore we head for a point on the central path, given by the path parameter

$$\mu = \frac{1}{2n}\mathrm{Tr}ZX.$$

Applying a Newton step to $F_\mu(Xy, Z) = 0$ at (X, y, Z), with μ as above, leads to

$$A(\Delta X) = 0 \tag{14}$$

$$\Delta Z = A^T(\Delta y) \tag{15}$$

$$Z(\Delta X) + (\Delta Z)X = \mu I - ZX. \tag{16}$$

The second equation can be used to eliminate ΔZ, the last to eliminate ΔX:

$$\Delta X = \mu Z^{-1} - X - Z^{-1}A^T(\Delta y)X.$$

Substituting this into the first equation gives the following linear system for Δy.

$$A(Z^{-1}A^T(\Delta y)X) = \mu A(Z^{-1}) - b.$$

This system is positive definite, see [90], and can therefore be solved quite efficiently by standard methods, yielding Δy. Backsubstitution gives ΔZ, which is symmetric, and ΔX, which need not be. Taking the symmetric part of ΔX gives the following new point (X^+, y^+, Z^+):

$$X^+ = X + t\frac{1}{2}(\Delta X + \Delta X^T)$$

$$y^+ = y + t\Delta y$$

$$Z^+ = Z + t\Delta Z.$$

The stepsize $t > 0$ is chosen so that $X^+ \succ 0,\ Z^+ \succ 0$. In practise one starts with $t = 1$ (full Newton step), and backtracks by multiplying the current t with a factor smaller than 1, such as 0.8, until positive definiteness of X^+ and Z^+ holds.

A theoretical convergence analysis shows the following. Let a small scalar $\epsilon > 0$ be given. If the path parameter μ to start a new iteration is chosen properly, then the full step ($t = 1$ above) is feasible in each iteration, and a primal feasible solution X and a dual feasible solution y, whose duality gap $b^Ty - \mathrm{Tr}(CX)$ is less than ϵ, can be found after $O(\sqrt{n}|\log \epsilon|)$ iterations; see the handbook [194], chapter 10.

2.3 Complexity

We consider here complexity issues for semidefinite programming. We saw above that for semidefinite programs satisfying the Slater constraint qualification, the primal problem (3) and its dual (4) can be solved in polynomial time to any fixed prescribed precision using interior point methods.

However, even if all input data $A_1, \ldots, A_m, C, b$ are rational valued, no polynomial bound has been established for the bitlengths of the intermediate numbers occurring in interior point algorithms. Therefore, interior point algorithms for semidefinite programming are shown to be polynomial in the real number model only, not in the bit number model of computation.

As a matter of fact, there are semidefinite programs with no rational optimum solution. For instance, the matrix $\begin{pmatrix} 1 & x \\ x & 2 \end{pmatrix} \oplus \begin{pmatrix} 2x & 2 \\ 2 & x \end{pmatrix}$ is positive semidefinite if and only if $x = \sqrt{2}$. (Given two matrices A, B, $A \oplus B$ denotes the matrix $\begin{pmatrix} A & 0 \\ 0 & B \end{pmatrix}$.) This contrasts with the situation of linear programming, where every rational linear program has a rational optimal solution whose bitlength is polynomially bounded in terms of the bit lengths of the input data (see [175]).

Another 'pathological' situation which may occur in semidefinite programming is that all feasible solutions are doubly exponential. Consider, for instance, the matrix (taken from [169]): $Q(x) := Q_1(x) \oplus \ldots \oplus Q_n(x)$, where $Q_1(x) := (x_1 - 2)$ and $Q_i(x) := \begin{pmatrix} 1 & x_{i-1} \\ x_{i-1} & x_i \end{pmatrix}$ for $i = 2, \ldots, n$. Then, $Q(x) \succeq 0$ if and only if $Q_i(x) \succeq 0$ for all $i = 1, \ldots, n$ which implies that $x_i \geq 2^{2^{i}-1}$ for $i = 1, \ldots, n$. Therefore, every rational feasible solution has exponential bitlength.

Semidefinite programs can be solved in polynomial time to an arbitrary prescribed precision in the bit model using the ellipsoid method (see [80]). More precisely, let K denote the set of feasible solutions to (3) and, given $\epsilon > 0$, set $S(K, \epsilon) := \{Y \mid \exists X \in K \text{ with } \|X - Y\| < \epsilon\}$ ('the points that are in the ϵ-neighborhood of K') and $S(K, -\epsilon) := \{X \in K \mid \|X - Y\| > \epsilon \text{ for all } Y \notin K\}$ ('the points in K that are at distance at least ϵ from the border of K'). Let L denote the maximum bit size of the entries of the matrices $A_1, \ldots, A_m$ and the vector b and assume that there is a constant $R > 0$ such that $\exists X \in K$ with $\|X\| \leq R$ if $K \neq \emptyset$. Then, the ellipsoid based algorithm, given $\epsilon > 0$, either finds $X \in S(K, \epsilon)$ for which $\text{Tr}(CY) \leq \text{Tr}(CX) + \epsilon$ for all $Y \in S(K, -\epsilon)$, or asserts that $S(K, -\epsilon) = \emptyset$. Its running time is polynomial in n, m, L, and $\log \epsilon$.

One of the fundamental open problems in semidefinite programming is the complexity of the following *semidefinite programming feasibility problem*[1] (F):

Given integral $n \times n$ symmetric matrices $Q_0, Q_1, \ldots, Q_m$, decide whether there exist real numbers $x_1, \ldots, x_m$ such that $Q_0 + x_1 Q_1 + \ldots + x_m Q_m \succeq 0$.

This problem belongs obviously to NP in the real number model (since one can test whether a matrix is positive semidefinite in polynomial time using Gaussian elimination), but it is not known whether it belongs to NP in the bit model of computation. Ramana [169] shows that problem (F) belongs to co-NP in the real number model, and that (F) belongs to NP if and only if it belongs to co-NP in the bit model. These two results are based on an extended exact duality theory for semidefinite programming. Namely, given a semidefinite program (P), Ramana [169] defines another semidefinite program (D) whose number of variables and coefficients bitlengths are polynomial in terms of the size of data in (P) and with the property that (P) is feasible if and only if (D) is infeasible.

Porkolab and Khachiyan [164] show that problem (F) can be solved in polynomial time (in the bit model) for *fixed* n or m. (More precisely, problem (F) can be solved in $O(mn^4) + n^{O(\min(m,n^2))}$ arithmetic operations over $Ln^{O(\min(m,n^2))}$-bit numbers, where L is the maximum bitlength of the entries of matrices $Q_0, \ldots, Q_m$.) Moreover, for any *fixed* m, one can decide in polynomial time (in the bit model) whether there exist *rational* numbers $x_1, \ldots, x_m$ such that $Q_0 + x_1 Q_1 + \ldots + x_m Q_m \succeq 0$ ([104]); this extends Lenstra's result ([128]) about polynomial time solvability of integer linear programming in fixed dimension to semidefinite programming. More generally, given a convex semi-algebraic set $K \subseteq \mathbb{R}^n$, one can find in polynomial time an integral point in K (if some exists) for any fixed dimension n [105]. When all the polynomials defining K are quadratic, this result still holds without the convexity assumption [25].

[1] The following is an equivalent form for the feasibility region of a semidefinite program (3). Indeed, a matrix X is of the form $Q_0 + \sum_{i=1}^{m} x_i Q_i$ if and only if it satisfies the system: $\text{Tr}(A_j X) = b_j$ $(j = 1, \ldots, p)$, where $A_1, \ldots, A_p$ span the orthogonal complement of the subspace of $\mathcal{S}_n$ generated by $Q_1, \ldots, Q_m$ and $b_j = \text{Tr}(A_j Q_0)$ for $j = 1, \ldots, p$.

A special instance of the semidefinite programming feasibility problem is the *semidefinite matrix completion problem* (MC), which consists of deciding whether a partially specified matrix can be completed to a positive semidefinite matrix. The complexity of problem (MC) is not known in general, even not for the class of partial matrices whose entries are specified on the main diagonal and on the positions corresponding to the edge set of a circuit. However, for circuits (and, more generally, for graphs with no K_4-minor), problem (MC) is known to be polynomial-time solvable in the real number model [121]. In the bit model, problem (MC) is known to be polynomial time solvable when the graph corresponding to the positions of the specified entries is chordal or can be made chordal by adding a *fixed* number of edges [121]. A crucial tool is a result of [79] asserting that a partial matrix A whose entries are specified on the edge set of a chordal graph can be completed to a positive semidefinite matrix if and only if every fully specified principal submatrix of A is positive semidefinite.

As mentioned above, one of the difficulties in the complexity analysis of semidefinite programming is the possible nonexistence of rational solutions. However, in the special case of the matrix completion problem, no example is known of a rational partial matrix having only irrational positive semidefinite completions. (Obviously, a rational completion exists if a positive *definite* completion exists.)

Further conditions are known for existence of positive semidefinite matrix completions, involving cut and metric polyhedra (see [118]); see the surveys [96], [120] for more information. In practice, positive semidefinite matrix completions can be computed using, e.g., the interior point algorithm of Johnson et al. [97]. This algorithm solves the problem:

$$\min f(X) \quad \text{subject to} \quad X \succeq 0,$$

where $f(X) := \sum_{i,j=1}^{n} (h_{ij})^2 (x_{ij} - a_{ij})^2$. Here H is a given nonnegative symmetric matrix with a positive diagonal and A is a given symmetric matrix corresponding to the partial matrix to be completed; the condition $h_{ij} = 0$ means that entry x_{ij} is free while $h_{ij} > 0$ puts a weight on forcing entry x_{ij} to be as close as possible to a_{ij}. The optimum value of the above program is equal to 0 precisely when there is a positive semidefinite matrix completion of A, where the entries of A corresponding to $h_{ij} = 0$ are unspecified.

2.4 Geometry

We discuss here some geometric properties of semidefinite programming. We refer to Chapter 3 in [194] for a detailed treatment. Let

$$K := \{X \in \mathrm{PSD}_n \mid \mathrm{Tr}(A_i X) = b_i \text{ for } i = 1, \ldots, m\}$$

denote the feasible region of a semidefinite program, where $A_1, \ldots, A_m \in \mathcal{S}_n$ and $b \in \mathbb{R}^m$. The set K is a convex set (called a *spectrahedron* in [170]) which inherits several of the geometric properties of the positive semidefinite cone PSD_n, in particular, concerning the structure of its faces. Recall that a set $F \subseteq K$ is a *face* of K if $X, Y \in F$ and $Z := \alpha X + (1 - \alpha) Y \in K$ for some $0 < \alpha < 1$ implies that $Z \in F$. Given $A \in K$, $F_K(A)$ denotes the smallest face of K containing A. A point $A \in K$ is an *extreme point* if $F_K(A) = \{A\}$. It is well known (see [92]) that, given a matrix $A \in \mathrm{PSD}_n$, the smallest face $F_{\mathrm{PSD}}(A)$ of PSD_n that contains A is given by

$$F_{\mathrm{PSD}}(A) = \{X \in \mathrm{PSD}_n \mid \ker A \subseteq \ker X\}.$$

(For a matrix X, $\ker X := \{x \in \mathbb{R}^n \mid Xx = 0\}$.) Hence, if A has rank r, then $F_{\mathrm{PSD}}(A)$ is isomorphic to the cone PSD_r and thus has dimension $\binom{r+1}{2}$. As K is the intersection of PSD_n with the affine space

$$\mathcal{A} := \{X \in \mathcal{S}_n \mid \mathrm{Tr}(A_i X) = b_i \text{ for } i = 1, \ldots, m\},$$

the face $F_K(A)$ for $A \in K$ is given by

$$F_K(A) = F_{\mathrm{PSD}}(A) \cap \mathcal{A} = \{X \in K \mid \ker A \subseteq \ker X\}.$$

One can compute the dimension of faces of K in the following manner (see Chapter 31.5 in [57]).

Let r denote the rank of A and let $A = QQ^T$, where Q is a $n \times r$ matrix of rank r. A matrix $B \in \mathcal{S}_n$ is called a *perturbation* of A if $A \pm tB \in K$ for some small $t > 0$. One can verify that B is a perturbation of A if and only if $B = QRQ^T$ for some matrix $R \in \mathcal{S}_r$ satisfying $\mathrm{Tr}(RQ^T A_i Q) = 0$ for all $i = 1, \ldots, m$. Then the dimension of $F_K(A)$ is equal to the rank of the set of perturbations of A and, therefore,

$$\dim\ F_K(A) = \binom{r+1}{2} - \mathrm{rank}\{Q^T A_i Q \mid i = 1, \ldots, m\}.$$

This implies:

$$A \text{ is an extreme point of } K \Longleftrightarrow \binom{r+1}{2} = \mathrm{rank}\{Q^T A_i Q \mid i = 1, \ldots, m\}. \tag{17}$$

We will use semidefinite programs as relaxations for 0/1 polytopes associated to combinatorial optimization problems; often the rank one matrices in the feasible region K correspond to the integer solutions of the combinatorial problem at hand. With this in mind, it is desirable to find a matrix $A \in K$ optimizing a given linear objective function over K having smallest possible rank. The smallest possible ranks are obviously achieved at extremal matrices of K. Some results have been obtained along these lines which we now mention.

As an application of (17), we have that if $K \neq \emptyset$ and $\mathrm{rank}\{A_i \mid i = 1, \ldots, m\} < \binom{r+2}{2}$, then there exists a matrix $X \in K$ with rank $X \leq r$ ([26], [158]). In fact, every extremal matrix X of K has this property; we will see below how to construct extremal matrices.

Barvinok [27] shows the following refinement. Suppose that K is a nonempty bounded set and that $\mathrm{rank}\{A_i \mid i = 1, \ldots, m\} = \binom{r+2}{2}$ for some $1 \leq r \leq n-2$, then there exists a matrix $X \in K$ with rank $X \leq r$. Barvinok's proof is nonconstructive and it is an open question how to find efficiently such X.

Barvinok [26] suggests the following approach for finding an extremal matrix in K. Let $C \in \mathcal{S}_n$ be a positive definite matrix and let $A \in K$ minimizing $\mathrm{Tr}(CX)$ over K. Barvinok shows that if C is sufficiently generic then A is an extremal point of K.

The following algorithm for constructing an extreme point of K has been suggested by several authors (see [3], [158]). Suppose we want to minimize the objective function $\mathrm{Tr}(CX)$ over K and assume that the minimum is finite. Given $A \in K$, the algorithm will construct an extremal matrix $A' \in K$ with objective value $\mathrm{Tr}(CA') \leq \mathrm{Tr}(CA)$. Using (17), one can verify whether A is an extreme point of K. If yes, then stop and return $A' = A$. Otherwise, one can find a nonzero matrix R belonging to the orthogonal complement in $\mathcal{S}_r$ of the space spanned by $Q^T A_i Q$ $(i = 1, \ldots, m)$; then $B := QRQ^T$ is a perturbation of A. If $\mathrm{Tr}(CB) > 0$ then replace B by $-B$. Let t be the largest possible scalar for which $A + tB \succeq 0$. Then, $A + tB$ belongs to the boundary of the face $F_K(A)$ and thus the face $F_K(A + tB)$ is strictly contained in $F_K(A)$. We iterate with $A + tB$ in place of A. In at most n iterations the algorithm returns an extreme point of K.

We conclude with some examples.

The Max-Cut spectrahedron. The following spectrahedron

$$\mathcal{E}_n := \{X \in \mathrm{PSD}_n \mid X_{ii} = 1 \ \forall i = 1, \dots, n\}$$

underlies the semidefinite relaxation for Max-Cut and will be treated in detail in Section 5. Its geometric properties have been investigated in [126, 127]. In particular, it is shown there that the only vertices (that is, the extreme points having a full dimensional normal cone) of $\mathcal{E}_n$ are its rank one matrices (corresponding to the cuts, i.e., the combinatorial objects in which we are interested). The spectrum of possible dimensions for the faces of $\mathcal{E}_n$ is shown to be equal to

$$\left[0, \binom{k_n}{2}\right] \cup \bigcup_{r=k_n+1}^{n} \left[\binom{r+1}{2} - n, \binom{r}{2}\right],$$

where $k_n := \lfloor \frac{n}{2} \rfloor + 1$. Moreover it is shown that the possible dimensions for the polyhedral faces of $\mathcal{E}_n$ are all integers k satisfying $\binom{k+1}{2} \le n$. Geometric properties of other tighter spectrahedra for max-cut are studied in [11], [125].

Sum of largest eigenvalues. We introduced in Example 4 two programs (5) and (6) permitting to express the sum of the k largest eigenvalues of a symmetric matrix. Let K and $\mathcal{Y}$ denote their respective feasible regions; that is,

$$K := \{X \in \mathcal{S}_n \mid I \succeq X \succeq 0, \ \mathrm{Tr}(X) = k\},$$

$$\mathcal{Y} := \{YY^T \mid Y \in \mathbb{R}^{n \times k} \text{ with } Y^T Y = I_k\}.$$

Lemma 7. *The extreme points of the set K are the matrices of $\mathcal{Y}$. Therefore, K is equal to the convex hull of the set $\mathcal{Y}$.*

Proof. Let X be an extreme point of K. Then all its eigenvalues belong to the segment $[0, 1]$. As $\mathrm{Tr}(X) = k$, it follows that X has at least k nonzero eigenvalues and thus $\mathrm{rank}(X) \ge k$. In fact, $\mathrm{rank}(X) = k$ since X is an extreme point of K. Now this implies that the only nonzero eigenvalue of X is 1 with multiplicity k and thus $X \in \mathcal{Y}$. Conversely, every matrix of $\mathcal{Y}$ is obviously an extreme point of K. ▮

Note the resemblance of the above result to the Birkhoff-König theorem asserting that the set of stochastic matrices is equal to the convex hull of the set of permutation matrices.

Euclidean distance matrix completions. Let $G = (V, E; d)$ be a weighted graph with $V = \{1, \dots, n\}$ and nonnegative edge weights $d \in \mathbb{Q}_+^E$. Given an integer r, we say that G is *r-realizable* if there exist points $v_1, \dots, v_n \in \mathbb{R}^r$ such that $d_{ij} = \|v_i - v_j\|$ for all edges $ij \in E$; G is said to be *realizable* if it r-realizable for some r. The problem of testing existence of a realization is known as the *Euclidean distance matrix completion problem* (EDM) (cf. [120] and Chapter 18 in [194] for surveys). It has important applications, e.g., to molecular conformation problems in chemistry and distance geometry (see [50]). As is well known, problem (EDM) can be formulated as a semidefinite programming problem. Namely, G is realizable if and only if the system:

$$X \succeq 0, \ X_{ii} + X_{jj} - 2X_{ij} = (d_{ij})^2 \ \text{ for } ij \in E \tag{18}$$

is feasible; moreover G is r-realizable if and only if the system (18) has a solution X with rank $X \leq r$. It follows from the above mentioned results about ranks of extremal points that if G is realizable, then G is r-realizable for some r satisfying $\binom{r+1}{2} \leq |E|$. Such a realization can be found using the above mentioned algorithm for finding extremal points (see [3], [26]).

It is also well known that the Euclidean distance matrix completion problem can be recast in terms of the positive semidefinite matrix completion problem (MC) treated earlier in Section 2.3 (see [119] for details). As a consequence, the complexity results mentioned earlier for problem (MC) also hold for problem (EDM). Namely, problem (EDM) can be solved in polynomial time in the bit number model when G can be made chordal by adding fixed number of edges, and (EDM) can be solved in polynomial time in the real number model when G has no K_4-minor [121].

An interior point algorithm is proposed in [4] for computing graph realizations. Alfakih [1, 2] studies rigidity properties of graph realizations in terms of geometric properties of certain associated spectrahedra.

When the graph G is not realizable, one can look for the smallest distortion needed to be applied to the edge weights in order to ensure existence of a realization. Namely, define this smallest distortion as the smallest scalar C for which there exist points $v_1, \ldots, v_n \in \mathbb{R}^n$ satisfying

$$\frac{1}{C} d_{ij} \leq \|v_i - v_j\| \leq d_{ij}$$

for all $ij \in E$. The smallest distortion can be computed using semidefinite programming. Bourgain [38] has shown that $C = O(\log n)$ if $G = K_n$ and d satisfies the triangle inequalities: $d_{ij} \leq d_{ik} + d_{jk}$ for all $i, j, k \in V$ (see also Chapter 10 in [57]). Since then research has been done for evaluating the minimum distortion for several classes of metric spaces including graph metrics (that is, when d is the path metric of graph G); see in particular [129], [130], [131].

3 Semidefinite Programming Versus Integer 0/1 Programming

3.1 A general paradigm

Suppose we want to solve a 0/1 linear programming problem:

$$\max\ c^T x \ \text{ subject to } \ Ax \le b,\ x \in \{0,1\}^n. \tag{19}$$

The classic polyhedral approach to this problem consists of formulating (19) as a linear programming problem:

$$\max\ c^T x \ \text{ subject to } \ x \in P$$

over the polytope

$$P := \text{conv}(\{x \in \{0,1\}^n \mid Ax \le b\})$$

and of applying to it linear programming techniques. For this one has to find the linear description of P or, at least, good linear relaxations of P. An initial linear relaxation of P is

$$K := \{x \in \mathbb{R}^n \mid Ax \le b\}$$

and, if $K \neq P$, one has to find 'cutting planes' permitting to strengthen the relaxation K by cutting off its fractional vertices. Extensive research has been done for finding (partial) linear descriptions for many polyhedra arising from specific combinatorial optimization problems by exploiting the combinatorial structure of the problem at hand. Next to that, research has also focused on developing general purpose methods applying to arbitrary 0/1 problems (or, more generally, integer programming problems).

An early such method, developed in the sixties by Gomory and based on integer rounding, permits to generate the so-called Chvátal-Gomory cuts. This class of cutting planes was later extended, in particular, by Balas [17] who introduced the disjunctive cuts. In the nineties several authors investigated lift-and-project methods for constructing cutting planes; the basic idea being of trying to represent a 0/1 polytope as projection of a polytope lying in higher dimension. These methods aim at constructing good *linear* relaxations of a given 0/1 polytope; all with the exception of the lift-and-project method of Lovász and Schrijver which permits, moreover, to construct *semidefinite* relaxations.

This idea of constructing semidefinite relaxations for a combinatorial problem goes back to the seminal work of Lovász [134] who introduced the semidefinite bound $\vartheta(G)$ for the stability number of a graph G, obtained by optimizing over a semidefinite relaxation TH(G) of the stable set polytope. An important application is the polynomial time solvability of the maximum stable set problem in perfect graphs. This idea was later again used successfully by Goemans and Williamson [77] who, using a semidefinite relaxation of the cut polytope, could prove an approximation algorithm with a good performance guarantee for the max-cut problem. Since then semidefinite programming has been widely used for approximating a variety of combinatorial optimization problems. This will be discussed in detail in further sections of this chapter.

For now we want to go back to the basic question of how to embed the 0/1 linear problem (19) in a semidefinite framework. A natural way of involving positive semidefiniteness is to introduce the matrix variable

$$Y = \binom{1}{x}(1\ x^T).$$

Then Y can be constrained to satisfy

$$\text{(i) } Y \succeq 0, \quad \text{(ii) } Y_{ii} = Y_{0i} \ \forall i = 1, \ldots, n.$$

Condition (ii) expresses the fact that $x_i^2 = x_i$ as $x_i \in \{0, 1\}$. One can write (i), (ii) equivalently as

$$Y = \begin{pmatrix} 1 & x^T \\ x & X \end{pmatrix} \succeq 0 \text{ where } x := \operatorname{diag}(X). \tag{20}$$

The objective function $c^T x$ can be modeled as $\langle \operatorname{diag}(c), X \rangle$. There are several possibilities for modeling a linear constraint $a^T x \leq \beta$ from the system $Ax \leq b$. The simplest way is to use the *diagonal representation*:

$$\langle \operatorname{diag}(a), X \rangle \leq \beta. \tag{21}$$

One can also replace $a^T x \leq \beta$ by its square $(\beta - a^T x)^2 \geq 0$, giving the inequality $(\beta \ - a^T) Y \binom{\beta}{-a} \geq 0$ which is however redundant under the assumption $Y \succeq 0$. Instead, when $a, \beta \geq 0$, one can use the *squared representation*: $(a^T x)^2 \leq \beta^2$; that is,

$$\langle aa^T, X \rangle \leq \beta^2 \tag{22}$$

or the *extended square representation*: $(a^T x)^2 \leq \beta \ (a^T x)$; that is,

$$\langle aa^T - \beta \ \operatorname{diag}(a), X \rangle \leq 0. \tag{23}$$

Another possibility is to exploit the fact that the variable x_i satisfies $0 \leq x_i \leq 1$ and to multiply $a^T x \leq \beta$ by x_i and $1 - x_i$, which yields the system:

$$\sum_{j=1}^{n} a_j X_{ij} \leq \beta X_{ii} \ (i = 1, \ldots, n), \quad \sum_{j=1}^{n} a_j (X_{jj} - X_{ij}) \leq \beta (1 - X_{ii}) \ (i = 1, \ldots, n). \tag{24}$$

One can easily compare the strengths of these various representations of the inequality $a^T x \leq \beta$ and verify that, if (20) holds, then

$$(24) \implies (23) \implies (22) \implies (21) \ .$$

Therefore, the constraints (24) define the strongest relaxation; they are, in fact, at the core of the lift-and-project methods by Lovász and Schrijver and by Sherali and Adams as we will see in Section 3.4. From an algorithmic point of view they are however the most expensive ones, as they involve $2n$ inequalities as opposed to one for the other relaxations. Helmberg et al. [91] made an experimental comparison of the various relaxations which seems to indicate that the best trade off between running time and quality is obtained when working with the squared representation.

Instead of treating each inequality of the system $Ax \leq b$ separately, one can also consider pairwise products of inequalities: $(\beta_i - a_i^T x) \cdot (\beta_j - a_j^T x) \geq 0$, yielding the inequalities: $(\beta_i \ - a_i^T) Y \binom{\beta_j}{-a_j} \geq 0$. This operation is also central to the lift-and-project methods as we will see later in this section.

3.2 Introduction on cutting planes and lift-and-project methods

Given a set $F \subseteq \{0,1\}^n$, we are interested in finding the linear description of the polytope $P := \text{conv}(F)$. A first (easy) step is to find a *linear programming formulation* for P; that is, to find a linear system $Ax \leq b$ for which the polytope $K := \{x \in \mathbb{R}^n \mid Ax \leq b\}$ satisfies $K \cap \{0,1\}^n = F$. If all vertices of K are integral, then $P = K$ and we are done. Otherwise we have to find cutting planes permitting to tighten the relaxation K and possibly find P after a finite number of iterations.

One of the first methods, which applies to general integral polyhedra, is the method of Gomory for constructing cutting planes. Given a linear inequality $\sum_i a_i x_i \leq \alpha$ valid for K where all the coefficients a_i are integers, the inequality $\sum_i a_i x_i \leq \lfloor \alpha \rfloor$ (known as a *Gomory-Chvátal cut*) is still valid for P but may eliminate some part of K. The *Chvátal closure* K' of K is defined as the solution set of all Chvátal-Gomory cuts; that is,

$$K' := \{x \in \mathbb{R}^n \mid u^T Ax \leq \lfloor u^T b \rfloor \text{ for all } u \geq 0 \text{ such that } u^T A \text{ integral}\}.$$

Then,

$$P \subseteq K' \subseteq K. \tag{25}$$

Set $K^{(1)} := K'$ and define recursively $K^{(t+1)} := (K^{(t)})'$. Chvátal [43] proved that K' is a polytope and that $K^{(t)} = \text{conv}(K)$ for some t; the smallest t for which this is true is the *Chvátal rank* of the polytope K. The Chvátal rank may be very large as it depends not only on the dimension n but also on the coefficients of the inequalities involved. However, when K is assumed to be contained in the cube $[0,1]^n$, its Chvátal rank is bounded by $O(n^2 \log n)$; if, moreover, $K \cap \{0,1\}^n = \emptyset$, then the Chvátal rank is at most n [34, 60]. Even if we can optimize a linear objective function over K in polynomial time, optimizing a linear objective function over the first Chvátal closure K' is a co-NP-hard problem in general [59].

Further classes of cutting planes have been investigated; in particular, the class of disjunctive cuts (studied in [17], and in [47] under the name of *split cuts*) which is known to be equivalent to Gomory's mixed integer cuts (cf. [48]). An inequality $a^T x \leq \alpha$ being a *disjunctive cut* for K if it is valid for the polytope $\text{conv}\Big((K \cap \{x \mid c^T x \leq c_0\}) \bigcup (K \cap \{x \mid c^T x \geq c_0 + 1\})\Big)$ for some integral $c \in \mathbb{Z}^n$, $c_0 \in \mathbb{Z}$. The *disjunctive closure* K' of K, defined as the solution set to all disjunctive cuts, is a polytope which satisfies again (25) [47]. One can iterate this operation of taking the disjunctive closure and it follows from results in [17] that P is found after n steps. However, optimizing over the first disjunctive closure is again a hard problem [39]. If we consider only the disjunctive cuts obtained from the disjunctions $x_j \leq 0$ and $x_j \geq 1$, then we obtain a tractable relaxation of K which coincides with the relaxation obtained in one iteration of the Balas-Ceria-Cornuéjols lift-and-project method (which will be described later in Section 3.4).

Another popular method is to try to represent P as the projection of another polytope Q lying in a higher (but preferably still polynomial) dimensional space. The idea behind being that the projection of a polytope Q may have more facets than Q itself. Hence it could be that even if P has an exponential number of facets, such Q exists having only a polynomial number of facets and lying in a space whose dimension is polynomial in the original dimension of P (such Q is then called a *compact representation* of P). If this is the case then we have a proof that any linear optimization problem over P can be solved in polynomial time. At this point let us stress that it is not difficult to find a lift Q of P with a simple structure and lying in a space of *exponential* dimension; indeed, as pointed out in Section 3.3, *any* n-dimensional 0/1 polytope can be realized as projection of a canonical simplex lying in the $(2^n - 1)$-space.

This idea of finding compact representations has been investigated for several polyhedra arising from combinatorial optimization problems; for instance, Barahona [20], Barahona and Mahjoub [23, 24], Ball, Liu and Pulleyblank [19], Maculan [138], Liu [132] have provided such representations for certain polyhedra related to Steiner trees, stable sets, metrics, etc. On the negative side, Yannakakis [195] proved that the matching polytope cannot have a compact representation satisfying a certain symmetry assumption.

Several general purpose methods have been developed for constructing projection representations for general 0/1 polyhedra; in particular, by Balas, Ceria and Cornuéjols [18] (the BCC method), by Sherali and Adams [178] (the SA method), by Lovász and Schrijver [136] (the LS method) and, recently, by Lasserre [116]. [These methods are also known under the following names: lift-and-project for BCC, Linearization-Reformulation Technique (RLT) for SA, and matrix-cuts for LS.] A common feature of these methods is the construction of a hierarchy

$$K \supseteq K_1 \supseteq K_2 \supseteq \ldots \supseteq K_n \supseteq P$$

of linear or semidefinite relaxations of P which finds the exact convex hull in n steps; that is, $K_n = P$. The methods also share the following important algorithmic property: If one can optimize a linear objective function over the initial relaxation K in polynomial time, then the same holds for the next relaxations K_t for any *fixed* t, when applying the BCC, SA or LS constructions; for the Lasserre construction, this is true under the more restrictive assumption that the matrix A has a polynomial number of rows.

The first three methods (BCC, SA and LS) provide three hierarchies of *linear* relaxations of P satisfying the following inclusions: the Sherali-Adams relaxation is contained in the Lovász-Schrijver relaxation which in turn is contained in the Balas-Ceria-Cornuéjols relaxation. All three can be described following a common recipe: Multiply each inequality of the system $Ax \leq b$ by certain products of the bound inequalities $x_i \geq 0$ and $1 - x_i \geq 0$, replace each square x_i^2 by x_i, and linearize the products $x_i x_j$ $(i \neq j)$ by introducing a new variable $y_{ij} = x_i x_j$. In this way, we obtain polyhedra in a higher dimensional space whose projection on the subspace $\mathbb{R}^n$ of the original x variable contains P and is contained in K. The three methods differ in the way of chosing the variables employed as multipliers and of iterating the basic step. The Lovász-Schrijver method can be strengthened by requiring positive semidefiniteness of the matrix (y_{ij}), which leads then to a hierarchy of positive semidefinite relaxations of P.

The construction of Lasserre produces a hierarchy of *semidefinite* relaxations of P which refines each of the above three hierarchies (BCC, SA and LS, even its positive semidefinite version). It was originally motivated by results about moment sequences and the dual theory of representation of nonnegative polynomials as sums of squares. It is however closely related to the SA method as both can be described in terms of requiring positive semidefiniteness of certain principal submatrices of the moment matrices of the problem.

We present in Section 3.3 some preliminary results which permit to show the convergence of the Lasserre and SA methods and to prove that every 0/1 polytope can be represented as the projection of a simplex in the $(2^n - 1)$-space. Then we describe in Section 3.4 the four lift-and-project methods and Sections 3.5 and 3.6 contain applications of these methods to the maximum stable set and cut problems. Section 3.7 presents extensions to (in general non convex) polynomial programming problems.

It will sometimes be convenient to view a polytope in $\mathbb{R}^n$ as being embedded in the hyperplane $x_0 = 1$ of $\mathbb{R}^{n+1}$. The following notation will be used throughout this paragraph. For a polytope P in

$\mathbb{R}^n$, its *homogenization*

$$\tilde{P} := \{\lambda\binom{1}{x} \mid x \in P,\ \lambda \geq 0\}$$

is a cone in $\mathbb{R}^{n+1}$ such that $P = \{x \in \mathbb{R}^n \mid \binom{1}{x} \in \tilde{P}\}$. For a cone C in $\mathbb{R}^n$,

$$C^* := \{y \in \mathbb{R}^n \mid x^T y \geq 0\ \forall x \in C\}$$

denotes its *dual cone.*

3.3 A canonical lifting construction

Let $\mathcal{P}(V) := 2^V$ denote the collection of all subsets of $V = \{1, \ldots, n\}$ and let Z be the square 0/1 matrix indexed by $\mathcal{P}(V)$ with entries

$$Z(I, J) = 1 \text{ if and only if } I \subseteq J.$$

As Z is upper triangular with ones on its main diagonal, it is nonsingular and its inverse Z^{-1} has entries

$$Z^{-1}(I, J) = (-1)^{|J \setminus I|} \text{ if } I \subseteq J,\ Z^{-1}(I, J) = 0 \text{ otherwise.}$$

For $J \subseteq V$, let Z^J denote the J-th column of Z. [The matrix Z is known as the *zeta matrix* of the lattice $\mathcal{P}(V)$ and the matrix Z^{-1} as its *Möbius matrix.*]

Given a subset $\mathcal{J} \subseteq \mathcal{P}(V)$, let $\mathcal{C}_{\mathcal{J}}$ denote the cone in $\mathbb{R}^{\mathcal{P}(V)}$ generated by the columns Z^J $(J \in \mathcal{J})$ of Z and let $P_{\mathcal{J}}$ be the 0/1 polytope in $\mathbb{R}^n$ defined as the convex hull of the incidence vectors of the sets in $\mathcal{J}$. Then $\mathcal{C}_{\mathcal{J}}$ is a simplicial cone,

$$\mathcal{C}_{\mathcal{J}} = \{y \in \mathbb{R}^{\mathcal{P}(V)} \mid Z^{-1}y \geq 0,\ (Z^{-1}y)_J = 0 \text{ for } J \in \mathcal{P}(V) \setminus \mathcal{J}\},$$

and $P_{\mathcal{J}}$ is the projection on $\mathbb{R}^n$ of the simplex $\mathcal{C}_{\mathcal{J}} \cap \{y \mid y_\emptyset = 1\}$. This shows therefore that any 0/1 polytope in $\mathbb{R}^n$ is the projection of a simplex lying in $\mathbb{R}^{2^n-1}$.

Given $y \in \mathbb{R}^{\mathcal{P}(V)}$, let $M_V(y)$ be the square matrix indexed by $\mathcal{P}(V)$ with entries

$$M_V(y)(I, J) := y(I \cup J) \tag{26}$$

for $I, J \subseteq V$; $M_V(y)$ is known as the *moment matrix* of the sequence y. (See Section 7.1 for motivation and further information.) As noted in [136], we have:

$$M_V(y) = Z\text{diag}(Z^{-1}y)Z^T.$$

Therefore, the cone $\mathcal{C}_{\mathcal{P}(V)}$ can be alternatively characterized by any of the following linear and positive semidefinite conditions:

$$y \in \mathcal{C}_{\mathcal{P}(V)} \Longleftrightarrow Z^{-1}y \geq 0 \Longleftrightarrow M_V(y) \succeq 0. \tag{27}$$

Suppose that $\mathcal{J}$ corresponds to the set of 0/1 solutions of a semi-algebraic system

$$g_\ell(x) \geq 0 \text{ for } \ell = 1, \ldots, m$$

where the g_ℓ's are polynomials in x. One can assume without loss of generality that each g_ℓ has degree at most one in every variable x_i and then one can identify g_ℓ with its sequence of coefficients indexed by $\mathcal{P}(V)$. Given $g, y \in \mathbb{R}^{\mathcal{P}(V)}$, define $g * y \in \mathbb{R}^{\mathcal{P}(V)}$ by

$$g * y := M(y)g; \text{ that is, } g * y(J) := \sum_I g_I y_{I \cup J} \text{ for } J \subseteq V. \tag{28}$$

It is noted in [124] that the cone $\mathcal{C}_{\mathcal{J}}$ can be alternatively characterized by the following positive semidefinite conditions:

$$y \in \mathcal{C}_{\mathcal{J}} \Longleftrightarrow M_V(y) \succeq 0 \text{ and } M_V(g_\ell * y) \succeq 0 \text{ for } \ell = 1, \ldots, m. \tag{29}$$

This holds, in particular, when $\mathcal{J}$ corresponds to the set of 0/1 solutions of a linear system $Ax \leq b$, i.e., in the case when each polynomial g_ℓ has degree 1.

3.4 The Balas-Ceria-Cornuéjols, Lovász-Schrijver, Sherali-Adams, and Lasserre methods

As before, $K = \{x \in [0,1]^n \mid Ax \leq b\}$ and $P = \text{conv}(K \cap \{0,1\}^n)$ is the 0/1 polytope whose linear description is to be found. It is convenient to assume that the bound constraints $0 \leq x_i \leq 1$ ($i \in \{1, \ldots, n\}$) are explicitely present in the linear description of K; let us rewrite the two systems $Ax \leq b$ and $0 \leq x_i \leq 1$ ($i \in \{1, \ldots, n\}$) as $\tilde{A}x \leq \tilde{b}$ and let m denote the number of rows of A.

The Balas-Ceria-Cornuéjols construction. Fix an index $j \in \{1, \ldots, n\}$. Multiply the system $\tilde{A}x \leq \tilde{b}$ by x_j and $1-x_j$ to obtain the nonlinear system: $x_j(\tilde{A}x - \tilde{b}) \leq 0$, $(1-x_j)(\tilde{A}x - \tilde{b}) \leq 0$. Replace x_j^2 by x_j and linearize by introducing new variables $y_i = x_i x_j$ ($i = 1, \ldots, n$); thus $y_j = x_j$. This defines a polytope in the (x,y)-space defined by $2(m+2n)$ inequalities: $\tilde{A}y - \tilde{b}x_j \leq 0$, $\tilde{A}(x-y) - \tilde{b}(1-x_j) \leq 0$. Its projection $P_j(K)$ on the subspace $\mathbb{R}^n$ indexed by the original x-variable satisfies

$$P \subseteq P_j(K) \subseteq K$$

and it can be verified that $P_j(K) = \text{conv}(K \cap \{x \mid x_j = 0, 1\})$. Iterate by defining $P_{j_1 \ldots j_t}(K) := P_{j_t}(P_{j_{t-1}} \ldots (P_{j_1}(K)) \ldots)$. Then,

$$P \subseteq P_{j_1 \ldots j_n}(K) \subseteq P_{j_1 \ldots j_{n-1}}(K) \subseteq \ldots \subseteq P_{j_1}(K) \subseteq K$$

and $P = P_{j_1 \ldots j_n}(K)$.

The Sherali-Adams construction. The first step is analogue to the first step of the BCC method except that we now multiply the system $\tilde{A}x \leq \tilde{b}$ by x_j and $1 - x_j$ for *all* indices $j \in \{1, \ldots, n\}$. More generally, for $t = 1, \ldots, n$, the t-th step goes as follows. Multiply the system $\tilde{A}x \leq \tilde{b}$ by each product $f_t(J_1, J_2) := \prod_{j \in J_1} x_j \cdot \prod_{j \in J_2} (1 - x_j)$ where J_1 and J_2 are disjoint subsets of V with $|J_1 \cup J_2| = t$. Replace each square x_i^2 by x_i and linearize each product $\prod_{i \in I} x_i$ by a new variable y_I. This defines a polytope $R_t(K)$ in the space of dimension $n + \binom{n}{2} + \ldots + \binom{n}{T}$ where $T := \min(t+1, n)$ (defined by $2^t \binom{n}{t}(m + 2n)$ inequalities) whose projection $S_t(K)$ on the subspace $\mathbb{R}^n$ of the original x-variable satisfies

$$P \subseteq S_n(K) \subseteq \ldots \subseteq S_{t+1}(K) \subseteq S_t(K) \subseteq \ldots \subseteq S_1(K) \subseteq K$$

and $P = S_n(K)$. The latter equality follows from facts in Section 3.3 as we now see.

Write the linear system $\tilde{A}x \leq \tilde{b}$ as $g_\ell^T \binom{1}{x} \geq 0$ ($\ell = 1, \ldots, m + 2n$) where $g_\ell \in \mathbb{R}^{n+1}$. Extend g_ℓ to a vector in $\mathbb{R}^{\mathcal{P}(V)}$ by adding zero coordinates. The linearization of the inequality $g_\ell^T \binom{1}{x} \cdot f_t(I, J) \geq 0$ reads:

$$\sum_{I \subseteq H \subseteq I \cup J} (-1)^{|H \setminus I|} g_\ell * y(H) \geq 0.$$

Using relation (27), one can verify that the set $R_t(K)$ can be alternatively described by the positive semidefinite conditions:

$$\begin{array}{l} M_U(g_\ell * y) \succeq 0 \ \text{ for } \ell = 1, \ldots, m \text{ and } U \subseteq V \text{ with } |U| = t, \\ M_U(y) \succeq 0 \text{ for } U \subseteq V \text{ with } |U| = t+1 \end{array} \tag{30}$$

(where $g_1, \ldots, g_m$ correspond to the system $Ax \leq b$). It then follows from (29) that the projection $S_n(K)$ of $R_n(K)$ is equal to P.

The Lovász-Schrijver construction. Let U be another linear relaxation of P which is also contained in the cube $Q := [0,1]^n$; write U as $\{x \in \mathbb{R}^n \mid u_r^T \binom{1}{x} \geq 0 \ \forall r = 1, \ldots, s\}$. Multiply each inequality $g_\ell^T \binom{1}{x} \geq 0$ by each inequality $u_r^T \binom{1}{x} \geq 0$ to obtain the nonlinear system: $u_r^T \binom{1}{x} \cdot g_\ell^T \binom{1}{x} \geq 0$ for all $\ell = 1, \ldots, m+2n$, $r = 1, \ldots, s$. Replace each x_i^2 by x_i and linearize by introducing a new matrix variable $Y = \binom{1}{x}(1 \ x^T)$. This defines the set $M(K,U)$ consisting of the symmetric matrices $Y = (y_{ij})_{i,j=0}^n$ satisfying

$$y_{jj} = y_{0j} \text{ for } j = 1, \ldots, n, \tag{31}$$

$$u_r^T Y g_\ell \geq 0 \text{ for all } r = 1, \ldots, s, \ \ell = 1, \ldots, m+2n \ \text{ [equivalently, } Y\tilde{U}^* \subseteq \tilde{K}]. \tag{32}$$

The first LS relaxation of P is defined as

$$N(K,U) := \{x \in \mathbb{R}^n \mid \binom{1}{x} = Ye_0 \text{ for some } Y \in M(K,U)\}.$$

Then, $P \subseteq N(K,U) \subseteq N(K,Q) \subseteq K$ and $N(K,K) \subseteq N(K,U)$ if $K \subseteq U$. One can obtain stronger relaxations by adding positive semidefiniteness. Let $M_+(K,U)$ denote the set of positive semidefinite matrices in $M(K,U)$ and $N_+(K,U) := \{x \in \mathbb{R}^n \mid \binom{1}{x} = Ye_0 \text{ for some } Y \in M_+(K,U)\}$. Then,

$$P \subseteq N_+(K,U) \subseteq N(K,U) \subseteq K.$$

The most extensively studied choice for U is $U := Q$, leading to the N operator. Set $N(K) := N(K,Q)$ and, for $t \geq 2$, $N^t(K) := N(N^{t-1}(K)) = N(N^{t-1}(K), Q)$. It follows from condition (32) that $N(K) \subseteq \text{conv}(K \cap \{x \mid x_j = 0,1\}) = P_j(K)$, the first BCC relaxation, and thus

$$N(K) \subseteq N_0(K) := \bigcap_{j=1}^n P_j(K). \tag{33}$$

[One can verify that $N_0(K)$ consists of the vectors $x \in \mathbb{R}^n$ for which $\binom{1}{x} = Ye_0$ for some matrix Y (not necessarily symmetric) satisfying (31) and (32) (with $U = Q$).] More generally, $N^t(K) \subseteq P_{j_1 \ldots j_t}(K)$ and, therefore, $P = N^n(K)$.

The choice $U := K$ leads to the stronger operator N', where we define $N'(K) := N(K,K)$ and, for $t \geq 2$,

$$(N')^t(K) := N((N')^{t-1}(K), K). \tag{34}$$

This operator is considered in [123] when applied to the cut polytope.

When using the relaxation $U = Q$, the first steps in the SA and LS constructions are identical; that is, $S_1(K) = N(K)$. The next steps are however distinct. A main difference between the two methods is that the LS procedure constructs the successive relaxations by a succession of t lift-and-project steps, each lifting taking place in a space of dimension $O(n^2)$, whereas the SA procedure carries out only one lifting step, occurring now in a space of dimension $O(n^{t+1})$; moreover, the projection step is not mandatory in the SA construction.

The Lasserre construction. We saw in relation (30) that the SA method can be interpreted as requiring positive semidefiniteness of certain principal submatrices of the moment matrices $M_V(y)$ and $M_V(g_\ell * y)$. The Lasserre method consists of requiring positive semidefiniteness of certain other principal matrices of those moment matrices. Namely, given an integer $t = 0, \ldots, n$, let $P_t(K)$ be defined by the conditions

$$M_{t+1}(y) \succeq 0, \ M_t(g_\ell * y) \succeq 0 \text{ for } \ell = 1, \ldots, m \tag{35}$$

and let $Q_t(K)$ denote the projection of $P_t(K)$ on $\mathbb{R}^n$. (For a vector $z \in \mathbb{R}^{\mathcal{P}(V)}$, $M_t(z)$ denotes the principal submatrix of $M_V(z)$ indexed by all sets $I \subseteq V$ with $|I| \leq t$.) Then,

$$P \subseteq Q_n(K) \subseteq Q_{n-1}(K) \subseteq \ldots \subseteq Q_1(K) \subseteq Q_0(K) \subseteq K$$

and it follows from (29) that $P = Q_n(K)$.

The construction of Lasserre [114, 116] was originally presented in terms of moment matrices indexed by integer sequences (rather than subsets of V) and his proof of convergence used results about moment theory and the representation of nonnegative polynomials as sums of squares. The presentation and the proof of convergence given here are taken from [124].

How do the four hierarchies of relaxations relate? The following inclusions hold among the relaxations $P_{j_1 \ldots j_t}(K)$ (BCC), $S_t(K)$ (SA), $N^t(K)$ and $N_+^t(K)$ (LS), and $Q_t(K)$ (Lasserre):

(i) $Q_1(K) \subseteq N_+(K) \subseteq Q_0(K)$

(ii) [136] For $t \geq 1$, $S_t(K) \subseteq N^t(K) \subseteq P_{j_1 \ldots j_t}(K)$

(iii) [124] For $t \geq 1$, $S_t(K) \subseteq N(S_{t-1}(K))$, $Q_t(K) \subseteq N_+(Q_{t-1}(K))$,
and thus $Q_t(K) \subseteq S_t(K) \cap N_+^t(K)$.

Summarizing, the Lasserre relaxation is the strongest among all four types of relaxations.

Algorithmic aspects. Efficient approximations to linear optimization problems over the 0/1 polytope P can be obtained by optimizing over its initial relaxation K or any of the stronger relaxations constructed using the BCC, LS, SA and Lasserre methods. Indeed, if one can optimize in polynomial time any linear objective function over K [equivalently (by the results in [80]), one can solve the separation problem for K in polynomial time], then, for any *fixed* t, the same holds for each of the relaxations $P_{j_1 \ldots j_t}(K)$, $S_t(K)$, $N^t(K)$, $N_+^t(K)$ in the BCC, SA, and LS hierarchies. This holds for the Lasserre relaxation $Q_t(K)$ under the more restrictive assumption that the linear system defining K has polynomial number of rows. Better approximations are obtained for higher values of t, at an increasing cost however. Computational experiments have been carried out using the various methods; see, in particular, [18], [41], [42] for results using the BCC method, [180] (and further references there) for results using the SA method, and to [52] for a computational study of the N_+ operator.

Worst case examples where n iterations are needed for finding P. Let us define the *rank* of K with respect to a certain lift-and-project method as the smallest number of iterations needed for finding P. Specifically, the N-rank of K is the smallest integer t for which $P = N^t(K)$; define similarly the N_+, N_0, SA and Lasserre ranks. We saw above that n is a common upper bound for any such rank.

As we will see in Section 3.5, the relaxation of the stable set polytope obtained with the Lovász-Schrijver N operator is much weaker than that obtained with the N_+-operator. For example, the fractionnal stable set polytope of K_n (defined by nonnegativity and the edge constraints) has N-rank $n-2$ while its N_+-rank is equal to 1! However, in the case of max-cut, no graph is known for which a similar result holds. Thus it is not clear in which situations the N_+-operator is significantly better, especially when applied iteratively. Some geometric results about the compared strengths of the N, N_+ and N_0 operators are given in [75]. As a matter of fact, there exist polytopes K having N_+-rank equal to n (thus, for them, adding positive semidefiniteness does not help!). As a first example, let

$$K := \left\{ x \in [0,1]^n \mid \sum_{i=1}^n x_i \geq \frac{1}{2} \right\};$$

then $P = \{x \in [0,1]^n \mid \sum_{i=1}^n x_i \geq 1\}$ and the Chvátal rank of K is therefore equal to 1. The N_+-rank of K is equal to n [46, 52] and its SA-rank as well [124]. As a second example, let

$$K := \left\{ x \in [0,1]^n \mid \sum_{i \in I} x_i + \sum_{i \notin I} (1-x_i) \geq \frac{1}{2} \ \forall I \subseteq \{1, \ldots, n\} \right\};$$

then $K \cap \{0,1\}^n = \emptyset$ and thus $P = \emptyset$. Then the N_+-rank of K is equal to n [46, 75] as well as its SA-rank [124]. In fact, the Chvátal rank of K is also equal to n [45]. The rank of K remains equal to n for the iterated operator N^* defined by $N^*(K) := N_+(K) \cap K'$, combining the Chvátal closure and the N_+-operator [46, 52]. The rank is also equal to n if in the definition of N^* we replace the Chvátal closure by the disjunctive closure [49].

General setting in which the four methods apply. We have described above how the various lift-and-project methods apply to 0/1 linear programs, i.e., to the case when K is a polytope and $P = \text{conv}(K \cap \{0,1\}^n)$. In fact, they apply in a more general context, still retaining the property that P is found after n steps. Namely, the Lovász-Schrijver method applies to the case when K and U are arbitrary convex sets, the condition (32) reading then $Y\tilde{U}^* \subseteq \tilde{K}$. The BCC and SA methods apply to mixed 0/1 linear programs [18, 179]. Finally, the Lasserre and Sherali-Adams methods apply to the case when K is a semi-algebraic set, i.e., when K is the solution set of a system of polynomial inequalities (since relation (29) holds in this context).

Moreover, various strengthenings of the basic SA method have been proposed involving, in particular, products of other inequalities than the bounds $0 \leq x_i \leq 1$ (cf., e.g., [41], [180], [181], [182]). A comparison between the Lasserre and SA methods for polynomial programming from the algebraic point of view of representations of positive polynomials is made in [117].

3.5 Application to the stable set problem

Given a graph $G = (V, E)$, a set $I \subseteq V$ is *stable* if no two nodes of I form an edge and the *stable set polytope* STAB(G) is the convex hull of the incidence vectors χ^S of all stable sets S of G, where $\chi_i^S = 1$ if $i \in S$ and $\chi_i^S = 0$ if $i \in V \setminus S$. As linear programming formulation for STAB(G), we consider the *fractional stable set polytope* FRAC(G) which is defined by the nonnegativity constraints: $x \geq 0$ and the *edge inequalities*:

$$x_i + x_j \leq 1 \text{ for } ij \in E. \tag{36}$$

Let us indicate how the various lift-and-project methods apply to the pair $P :=$ STAB(G), $K :=$ FRAC(G).

The LS relaxations $N(\mathrm{FRAC}(G))$ and $N_+(\mathrm{FRAC}(G))$ are studied in detail in [136] where the following results are shown. The polytope $N(\mathrm{FRAC}(G))$ is completely described by nonnegativity, the edge constraints (36) and the *odd hole inequalities*:

$$\sum_{i\in V(C)} x_i \leq \frac{|C|-1}{2} \text{ for } C \text{ odd circuit in } G. \tag{37}$$

Moreover, $N(\mathrm{FRAC}(G)) = N_0(\mathrm{FRAC}(G))$. Therefore, this gives a compact representation for the stable set polytope of t-perfect graphs (they are the graphs whose stable set polytope is completely determined by nonnegativity together with edge and odd hole constraints).

Other valid inequalities for $\mathrm{STAB}(G)$ include the *clique inequalities*:

$$\sum_{i\in Q} x_i \leq 1 \text{ for } Q \text{ clique in } G. \tag{38}$$

The smallest integer t for which (38) is valid for $N^t(\mathrm{FRAC}(G))$ is $t = |Q| - 2$ while (38) is valid for $N_+(\mathrm{FRAC}(G))$. Hence the N_+ operator yields a stronger relaxation of $\mathrm{STAB}(G)$ and equality $N_+(\mathrm{FRAC}(G)) = \mathrm{STAB}(G)$ holds for perfect graphs (they are the graphs for which $\mathrm{STAB}(G)$ is completely determined by nonnegativity and the clique inequalities; cf. Theorem 8). Odd antihole and odd wheel inequalities are also valid for $N_+(\mathrm{FRAC}(G))$.

Given a graph G on n nodes with stability number $\alpha(G)$ (i.e., the maximum size of a stable set in G), the following bounds hold for the N-rank t of $\mathrm{FRAC}(G)$ and its N_+-rank t_+:

$$\frac{n}{\alpha(G)} - 2 \leq t \leq n - \alpha(G) - 1, \; t_+ \leq \alpha(G).$$

The Sherali-Adams method does not seem to give a significant improvement, since the quantity $\frac{n}{\alpha(G)} - 2$ remains a lower bound for the SA-rank [124].

The Lasserre hierarchy refines the sequence $N_+^t(\mathrm{FRAC}(G))$. Indeed, it is shown in [124] that, for $t \geq 1$, the set $Q_t(\mathrm{FRAC}(G))$ can be alternatively described as the projection of the set

$$M_{t+1}(y) \succeq 0, \; y_{ij} = 0 \text{ for all edges } ij \in E, \; y_\emptyset = 1. \tag{39}$$

This implies that $Q_{\alpha(G)-1}(\mathrm{FRAC}(G)) = \mathrm{STAB}(G)$; that is, the Lasserre rank of $\mathrm{FRAC}(G)$ is at most $\alpha(G) - 1$. The inclusion $Q_{\alpha(G)-1}(\mathrm{FRAC}(G)) \subseteq N_+^{\alpha(G)-1}(\mathrm{FRAC}(G))$ is strict, for instance, when G is the line graph of K_n (n odd) since the N_+-rank of $\mathrm{FRAC}(G)$ is then equal to $\alpha(G)$ ([188]).

We conclude with a comparison with the basic semidefinite relaxation of $\mathrm{STAB}(G)$ by the theta body $\mathrm{TH}(G)$, which is defined by

$$\mathrm{TH}(G) := \{x \in \mathbb{R}^n \mid \binom{1}{x} = Ye_0 \text{ for some } Y \succeq 0 \text{ s.t. } Y_{ii} = Y_{0i} \; (i \in V), \; Y_{ij} = 0 \; (ij \in E)\}. \tag{40}$$

Comparing with (39), we see that $Q_t(\mathrm{FRAC}(G))$ ($t \geq 1$) is a natural generalization of the SDP relaxation $\mathrm{TH}(G)$ satisfying the following chain of inclusions:

$$Q_t(\mathrm{FRAC}(G)) \subseteq Q_1(\mathrm{FRAC}(G)) \subseteq N_+(\mathrm{FRAC}(G)) \subseteq \mathrm{TH}(G) \subseteq Q_0(\mathrm{FRAC}(G)).$$

Section 4.2 below contains a detailed treatment of the relaxation $\mathrm{TH}(G)$.

3.6 Application to the max-cut problem

We consider here how the various lift-and-project methods can be used for constructing relaxations of the cut polytope. Section 5 will focus on the most basic SDP relaxation of the cut polytope and, in particular, on how it can be used for designing good approximation algorithms for the max-cut problem. As is well known (cf. (68)), the max-cut problem can be formulated as an unconstrained quadratic ±1 problem:

$$\max\ x^T A x \ \text{ subject to } \ x \in \{\pm 1\}^n \tag{41}$$

for some (suitably defined) symmetric matrix A.

As we are now working with ±1 variables instead of 0/1 variables, one should appropriately modify some of the definitions given earlier in this section. For instance, the condition (31) in the definition of the LS matrix operator M now reads $y_{ii} = y_{00}$ for all $i \in \{1, \ldots, n\}$ (in place of $y_{ii} = y_{0i}$) and the (I, J)-th entry of the moment matrix $M_V(y)$ is now $y(I \Delta J)$ (instead of $y(I \cup J)$ as in (26)).

There are two possible strategies for constructing relaxations of the max-cut problem (41). The first possible strategy is to linearize the quadratic objective function, to formulate (41) as a linear problem

$$\max\ \langle A, X\rangle \ \text{ subject to } \ X \in \mathrm{CUT}_n$$

over the *cut polytope*

$$\mathrm{CUT}_n := \mathrm{conv}(xx^T \mid x \in \{\pm 1\}^n),$$

and to apply the various lift-and-project methods to some linear relaxation of $\mathrm{CUT}(K_n)$. As linear programming formulation for CUT_n, one can take the *metric polytope* MET_n which is defined as the set of symmetric matrices X with diagonal entries 1 satisfying the triangle inequalities:

$$X_{ij} + X_{ik} + X_{jk} \geq -1,\ X_{ij} - X_{ik} - X_{jk} \geq -1$$

for all distinct $i, j, k \in \{1, \ldots, n\}$.

Given a graph $G = (V, E)$ $(V = \{1, \ldots, n\})$, $\mathrm{CUT}(G)$ and $\mathrm{MET}(G)$ denote, respectively, the projections of CUT_n and MET_n on the subspace $\mathbb{R}^E$ indexed by the edge set of G. Barahona and Mahjoub [23] show that $\mathrm{CUT}(G) \subseteq \mathrm{MET}(G)$ with equality if and only if G has no K_5-minor. Laurent [123] studies how the Lovász-Schrijver construction applies to the pair $P := \mathrm{CUT}(G)$ and $K := \mathrm{MET}(G)$. The following results are shown there: Equality $N_0^t(\mathrm{MET}(G)) = \mathrm{CUT}(G)$ holds if G has a set of t edges whose contraction produces a graph with no K_5-minor (recall the definition of N_0 from (33)). In particular, $N^{n-\alpha(G)-3}(\mathrm{MET}(G)) = \mathrm{CUT}(G)$ if G has a maximum stable set whose deletion leaves at most three connected components and $N^{n-\alpha(G)-3}(G) = \mathrm{CUT}(G)$. Here, $N^t(G)$ denotes the projection on the subspace indexed by the edge set of G of the set $N^t(\mathrm{MET}(K_n))$. The inclusion $N^t(G) \subseteq N^t(\mathrm{MET}(G))$ holds obviously. Therefore, the N-rank of $\mathrm{MET}(K_n)$ is at most $n-4$, with equality for $n \leq 7$ (equality is conjectured for any n). A stronger relaxation is obtained when using the N' operator (recall the definition of N' from (34)). Indeed, $N'(\mathrm{MET}(K_6)) = \mathrm{CUT}(K_6) \subset N(\mathrm{MET}(K_6))$ and the N'-rank of $\mathrm{MET}(K_n)$ is at most $n-5$ for $n \geq 6$.

Another possible strategy is to apply the lift-and-project constructions to the set $K := [-1, 1]^n$ and to project on the subspace indexed by the set E_n of all pairs ij of points of V (instead of projecting on the space $\mathbb{R}^n$ indexed by the singletons of V). The SA and Lasserre methods converge now in $n-1$ steps (as there is no additional linear constraint beside the constraints expressing membership in the cube).

The t-th relaxation in the SA hierarchy is determined by all the inequalities valid for $\mathrm{CUT}(K_n)$ that are induced by at most $t+1$ points. Thus, the relaxation of order $t = 1$ is the cube $[-1,1]^E$ while the relaxation of order $t = 2$ is the metric polytope $\mathrm{MET}(K_n)$.

The t-th relaxation in the Lasserre hierarchy, denoted as $Q_t(G)$, is the projection on the subspace $\mathbb{R}^E$ indexed by the edge set of G of the set of vectors y satisfying

$$M_{t+1}(y) = (y_{I\Delta J})_{\substack{I,J\subseteq V\\ |I|,|J|\leq t+1}} \succeq 0,\ y_\emptyset = 1. \tag{42}$$

Equivalently, one can replace in (42) the matrix $M_{t+1}(y)$ by its principal submatrix indexed by the subsets whose cardinality has the same parity as $t+1$. Therefore, for $t = 0$, $Q_0(K_n)$ coincides with the basic semidefinite relaxation

$$\{X = (X_{ij})_{i,j=1}^n \mid X \succeq 0,\ X_{ii} = 1\ \forall i \in \{1,\ldots,n\}\}$$

of the cut polytope. For $t = 1$, $Q_1(K_n)$ consists of the vectors $x \in \mathbb{R}^{E_n}$ for which $\binom{1}{x} = Ye_0$ for some matrix $Y \succeq 0$ indexed by $\{\emptyset\} \cup E_n$ satisfying

$$Y_{ij,ik} = Y_{\emptyset,jk}, \tag{43}$$

$$Y_{ij,hk} = Y_{ih,jk} = Y_{ik,jh} \tag{44}$$

for all distinct $i,j,h,k \in \{1,\ldots,n\}$.

Using Lagrangian duality, Anjos and Wolkowicz [10] introduced the relaxation F_n of $\mathrm{CUT}(K_n)$ which is defined as the set of all $x \in \mathbb{R}^{E_n}$ for which $\binom{1}{x} = Ye_0$ for some $Y \succeq 0$ indexed by $\{\emptyset\} \cup E_n$ satisfying (43). Thus

$$Q_1(K_n) \subseteq F_n$$

(with strict inclusion if $n \geq 5$). It is interesting to note that the relaxation F_n is stronger than the basic linear relaxation by the metric polytope [10]; that is,

$$F_n \subseteq \mathrm{MET}(K_n).$$

Indeed, let $x \in F_n$ with $\binom{1}{x} = Ye_0$ for some $Y \succeq 0$ satisfying (43). The principal submatrix X of Y indexed by $\{\emptyset, 12, 13, 23\}$ has the form $\begin{array}{c|cccc} & \emptyset & 12 & 13 & 23 \\ \hline \emptyset & 1 & x_{12} & x_{13} & x_{23} \\ 12 & x_{12} & 1 & x_{23} & x_{13} \\ 13 & x_{13} & x_{23} & 1 & x_{12} \\ 23 & x_{23} & x_{13} & x_{12} & 1 \end{array}$. Now $e^TXe = 4(1+x_{12}+x_{13}+x_{23}) \geq 0$ implies one of the triangle inequalities for the triple (1,2,3); the other triangle inequalities follow by suitably flipping signs in X.

Laurent [125] shows that

$$Q_t(G) \subseteq N_+^{t-1}(G)$$

for any $t \geq 1$. Therefore, the second strategy seems to be the most attractive one. Indeed, the relaxation $Q_t(G)$ is at least as tight as $N_+^{t-1}(G)$ and moreover it has a simpler explicit description (given by (42)) while the set $N_+^{t-1}(G)$ has only a recursive definition. We refer to [125] for a detailed study of geometric properties of the set of (moment) matrices of the form (42).

3.7 Extensions to polynomial programming

Quadratic programming. Suppose we want to solve the program

$$p^* := \min \; g_0(x) \text{ subject to } g_\ell(x) \geq 0 \; (\ell = 1, \ldots, m) \tag{45}$$

where $g_0, g_1, \ldots, g_m$ are quadratic functions of the form: $g_\ell(x) = x^T Q_\ell x + 2q_\ell^T x + \gamma_\ell$ (Q_ℓ symmetric $n \times n$ matrix, $q_\ell \in \mathbb{R}^n$, $\gamma_\ell \in \mathbb{R}$). For any ℓ, define the matrix $P_\ell := \begin{pmatrix} \gamma_\ell & q_\ell^T \\ q_\ell & Q_\ell \end{pmatrix}$. Then, $g_\ell(x) = \langle P_\ell, \begin{pmatrix} 1 & x^T \\ x & xx^T \end{pmatrix} \rangle$. This suggests the following natural positive semidefinite relaxation of (45):

$$\min \langle P_0, Y \rangle \;\text{ subject to } Y \succeq 0, \; Y_{00} = 1, \; \langle P_\ell, Y \rangle \geq 0 \; (\ell = 1, \ldots, m). \tag{46}$$

Let $F := \{x \in \mathbb{R}^n \mid g_\ell(x) \geq 0 \; (\ell = 1, \ldots, m)\}$ denote the feasible set of (45) and

$$\hat{F} := \{x \in \mathbb{R}^n \mid \; \binom{1}{x} = Ye_0 \text{ for some } Y \succeq 0 \text{ satisfying } \langle P_\ell, Y \rangle \geq 0 \; \text{ for all } \ell = 1, \ldots, m\} \tag{47}$$

its natural semidefinite relaxation. It is shown in [71, 113] that $\hat{F}$ can be alternatively described by the following quadratic system:

$$\hat{F} = \{x \in \mathbb{R}^n \mid \; \textstyle\sum_{\ell=1}^m t_\ell g_\ell(x) \geq 0 \text{ for all } t_\ell \geq 0 \; \text{ for which } \textstyle\sum_{\ell=1}^m t_\ell Q_\ell \preceq 0\}. \tag{48}$$

If, in (47), one omits the condition $Y \succeq 0$ and, in (48), the condition $\sum_\ell t_\ell Q_\ell \preceq 0$ is replaced by $\sum_\ell t_\ell Q_\ell = 0$, then one obtains a linear relaxation $\hat{F}_L$ of F such that $\text{conv}(F) \subseteq \hat{F} \subseteq \hat{F}_L$.

Using this construction of linear/semidefinite relaxations, Kojima and Tunçel [113] construct a hierarchy of successive relaxations of F that converges asymptotically to $\text{conv}(F)$. Lasserre [115] also constructs such a hierarchy which applies, more generally, to polynomial programs; we expose it below.

Polynomial programming. Consider now the program (45) where all the g_ℓ's are polynomials in $x = (x_1, \ldots, x_n)$. Let w_ℓ be the degree of g_ℓ, $v_\ell := \lceil \frac{w_\ell}{2} \rceil$ and $v := \max_{\ell=1,\ldots,m} v_\ell$. We need some definitions.

Given a sequence $y = (y_\alpha)_{\alpha \in \mathbb{Z}_+^n}$ indexed by $\mathbb{Z}_+^n$, its *moment matrix* is

$$M^{\mathbb{Z}}(y) := (y_{\alpha+\beta})_{\alpha,\beta \in \mathbb{Z}_+^n} \tag{49}$$

and, given an integer $t \geq 0$, $M_t^{\mathbb{Z}}(y)$ is the principal submatrix of $M^{\mathbb{Z}}(y)$ indexed by the sequences $\alpha \in \mathbb{Z}_+^n$ with $|\alpha| := \sum_i \alpha_i \leq t$. [Note that the moment matrix $M_V(y)$ defined earlier in (26) corresponds to the principal submatrix of $M^{\mathbb{Z}}(y)$ indexed by the sequences $\alpha \in \{0,1\}^n$, after replacing y_α by $y_{\alpha'}$ where $\alpha_i' := \min(\alpha_i, 1)$ for all i.] The operation from (28) extends to sequences indexed by $\mathbb{Z}_+^n$ in the following way:

$$g, \; y \in \mathbb{R}^{\mathbb{Z}_+^n} \rightsquigarrow g * y := (\sum_\beta g_\beta y_{\alpha+\beta})_{\alpha \in \mathbb{Z}_+^n}. \tag{50}$$

Given $x \in \mathbb{R}^n$, define the sequence $y \in \mathbb{R}^{\mathbb{Z}_+^n}$ with α-th entry $y_\alpha := \prod_{i=1}^n x_i^{\alpha_i}$ for $\alpha \in \mathbb{Z}_+^n$. Then, $M_t^{\mathbb{Z}}(y) = yy^T \succeq 0$ (where we use the same symbol y for denoting the truncated vector $(y_\alpha)_{|\alpha| \leq t}$) and $M_t^{\mathbb{Z}}(g_\ell * y) = g_\ell(x) \cdot M_t^{\mathbb{Z}}(y) \succeq 0$ if $g_\ell(x) \geq 0$. This observation leads naturally to the following relaxations of the set F, introduced by Lasserre [115].

For $t \geq v-1$, let $\mathcal{Q}_t(F)$ be the convex set defined as the projection of the solution set to the system

$$M^{\mathbb{Z}}_{t+1}(y) \succeq 0, \;\; M^{\mathbb{Z}}_{t-v_\ell+1}(g_\ell * y) \succeq 0 \;\text{ for } \ell = 1, \ldots, m, \; y_0 = 1 \tag{51}$$

on the subspace $\mathbb{R}^n$ indexed by the variables y_α for $\alpha = (1, 0, \ldots, 0), \ldots, (0, \ldots, 0, 1)$ (identified with $x_1, \ldots, x_n$). Then,

$$\text{conv}(F) \subseteq \mathcal{Q}_{t+1}(F) \subseteq \mathcal{Q}_t(F).$$

Lasserre [115] shows that

$$\bigcap_{t \geq v-1} \mathcal{Q}_t(F) = \text{conv}(F);$$

that is, the hierarchy $(\mathcal{Q}_t(F))_t$ converges asymptotically to $\text{conv}(F)$. This equality holds under some technical assumption on F which holds, for instance, when F is the set of 0/1 solutions of a polynomial system and the constraints $x_i(1-x_i) = 0$ $(i \in \{1, \ldots, n\})$ are present in the description of F, or when the set $\{x \mid g_\ell(x) \geq 0\}$ is compact for at least one of the constraints defining F. Lasserre's result relies on a result about representations of positive polynomials as sums of squares, to which we will come back in Section 7.1.

In the quadratic case, when all g_ℓ are quadratic polynomials, one can verify that the first Lasserre relaxation $\mathcal{Q}_0(F)$ coincides with the basic SDP relaxation $\hat{F}$ defined in (47); that is,

$$\mathcal{Q}_0(F) = \hat{F}.$$

Consider now the 0/1 case when F is the set of 0/1 solutions of a polynomial system; write F as

$$F = \{x \in \mathbb{R}^n \mid g_\ell(x) \geq 0 \; (\ell = 1, \ldots, m), \; h_i(x) := x_i - x_i^2 = 0 \; (i = 1, \ldots, n)\}.$$

One can assume without loss of generality that each g_ℓ has degree at most 1 in every variable. The set

$$K := \{x \in [0,1]^n \mid g_\ell(x) \geq 0 \; (\ell = 1, \ldots, m)\}$$

is a natural relaxation of F. We have constructed in Section 3.4 the successive relaxations $Q_t(K)$ of $\text{conv}(F)$ satisfying $\text{conv}(F) = Q_{n+v-1}(K)$; their construction used moment matrices indexed by the subsets of V while the definition of $\mathcal{Q}_t(F)$ involves moment matrices indexed by integer sequences. However, the condition $M^{\mathbb{Z}}_t(h_i * y) = 0$ (present in the definition of $\mathcal{Q}_t(F)$) permits to show that the two definitions are equivalent; that is,

$$Q_t(K) = \mathcal{Q}_t(F) \;\text{ for } t \geq v-1.$$

See [124] for details.

In the quadratic 0/1 case, we find therefore that

$$\hat{F} = \mathcal{Q}_0(F) = Q_0(K).$$

As an example, given a graph $G = (V = \{1, \ldots, n\}, E)$, consider the set

$$F := \{x \in \{0,1\}^n \mid x_i x_j = 0 \;\text{ for all } ij \in E\};$$

then $\text{conv}(F)$ is equal to the stable set polytope of G. It follows from the definitions that $\hat{F}$ coincides with the basic SDP relaxation $\text{TH}(G)$ (defined in (40)). Therefore, $\mathcal{Q}_0(F) = \text{TH}(G)$ while the inclusion

$\mathrm{TH}(G) \subseteq Q_0(\mathrm{FRAC}(G))$ is strict in general. Hence one obtains stronger relaxations for the stable set polytope $\mathrm{STAB}(G)$ when starting from the above quadratic representation F for stable sets rather than from the linear relaxation $\mathrm{FRAC}(G)$. Applying the equivalent definition (48) for $\hat{F}$, one finds that

$$\mathrm{TH}(G) = \left\{ x \in \mathbb{R}^n \mid x^T M x - \sum_{i=1}^{n} M_{ii} x_i \leq 0 \text{ for } M \succeq 0 \text{ with } M_{ij} = 0 \ (i \neq j \in V, ij \notin E) \right\}. \qquad (52)$$

(This formulation of $\mathrm{TH}(G)$ also follows using the duality between the cone of completable partial positive semidefinite matrices and the cone of positive semidefinite matrices having zeros at the positions of unspecified entries; cf. [122].) See Section 4.2 for further information about the semidefinite relaxation $\mathrm{TH}(G)$.

4 Semidefinite Relaxation for the Maximum Stable Set Problem

Given a graph $G = (V, E)$, its *stability number* $\alpha(G)$ is the maximum cardinality of a stable set in G, and its *clique number* $\omega(G)$ is the maximum cardinality of a clique in G. Given an integer $k \geq 1$, a *k-colouring* of G is an assignment of numbers from $\{1, \ldots, k\}$ (*colours*) to the nodes of G in such a way that adjacent nodes receive distinct colours; in other words, a k-colouring is a partition of V into k stable sets. The *colouring number* (or *chromatic number*) $\chi(G)$ is the smallest integer k for which G has a k-colouring. With $\overline{G} = (V, \overline{E})$ denoting the complementary graph of G, the following holds trivially:

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

The inequality $\omega(G) \leq \chi(G)$ is strict, for instance, for odd circuits of length ≥ 5 and their complements. Berge [30] defined a graph G to be *perfect* if $\omega(G') = \chi(G')$ for every induced subgraph G' of G and he conjectured that a graph is perfect if and only if it does not contain a circuit of length ≥ 5 or its complement as an induced subgraph; this is the (still open) *strong perfect graph conjecture.* Lovász [133] proved that the complement of a perfect graph is again perfect, solving another conjecture of Berge. As we will see later in this section, perfect graphs can also be characterized in terms of integrality of certain associated polyhedra.

Computing the stability number or the chromatic number of a graph are hard problems; more precisely, given an integer k, it is an NP-complete problem to decide whether $\alpha(G) \geq k$ or $\chi(G) \leq k$ [103]. Deciding whether a graph is 2-colourable can be done in polynomial time (as this happens if and only if the graph is bipartite). On the other hand, while every planar graph is 4-colourable (by the celebrated four colour theorem), it is NP-complete to decide whether a planar graph is 3-colourable [73]. When restricted to the class of perfect graphs, the maximum stable set problem and the colouring problem can be solved in polynomial time. This result relies on the use of the Lovász theta function $\vartheta(G)$ which can be computed (with an arbitrary precision) in polynomial time (as the optimum of a semidefinite program) and satisfies the 'sandwich' inequalities:

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

The polynomial time solvability of the maximum stable set problem for perfect graphs is one of the first beautiful applications of semidefinite programming to combinatorial optimization and, up to today, no other purely combinatorial method is known for proving this.

4.1 The basic linear relaxation

As before, the *stable set polytope* STAB(G) is the polytope in $\mathbb{R}^V$ defined as the convex hull of the incidence vectors of the stable sets of G, FRAC(G) is its linear relaxation defined by nonnegativity and the edge inequalities (36), and QSTAB(G) denotes the linear relaxation of STAB(G) defined by nonnegativity and the clique inequalities (38). Therefore,

$$\text{STAB}(G) \subseteq \text{QSTAB}(G) \subseteq \text{FRAC}(G)$$

and

$$\alpha(G) = \max(e^T x \mid x \in \text{STAB}(G))$$

setting $e := (1, \ldots, 1)^T$. One can easily see that equality STAB$(G) =$ FRAC(G) holds if and only if G is a bipartite graph with no isolated nodes; thus the maximum stable set problem for bipartite graphs can be solved in polynomial time as a linear programming problem over FRAC(G). Fulkerson [72] and Chvátal [44] show:

Theorem 8. *A graph G is perfect if and only if* $\mathrm{STAB}(G) = \mathrm{QSTAB}(G)$.

This result does not (yet) help for computing efficiently $\alpha(G)$ for perfect graphs. Indeed, optimizing over the linear relaxation QSTAB(G) is, unfortunately, a hard problem in general (as hard as the original problem, since the membership problem for QSTAB(G) is nothing but a maximum weight clique problem in G.) Proving polynomiality requires the use of the semidefinite relaxation TH(G) as we see later in this section.

4.2 The theta function $\vartheta(G)$ and the basic semidefinite relaxation TH(G)

Lovász [134] introduced the following parameter $\vartheta(G)$, known as the *theta number*:

$$\begin{array}{rll} \vartheta(G) := & \max & e^T X e \\ & \text{s.t.} & \mathrm{Tr}(X) = 1 \\ & & X_{ij} = 0 \ (i \neq j, \ ij \in E) \\ & & X \succeq 0. \end{array} \tag{53}$$

The theta number has two important properties: it can be computed with an arbitrary precision in polynomial time (as optimum value of a semidefinite program) and it provides bounds for the stability and chromatic numbers. Namely,

$$\alpha(G) \leq \vartheta(G) \leq \chi(\overline{G}). \tag{54}$$

To see that $\alpha(G) \leq \vartheta(G)$, consider a maximum stable set S; then the matrix $X := \frac{1}{|S|}\chi^S(\chi^S)^T$ is feasible for the program (53) and $\alpha(G) = e^T X e$. To see that $\vartheta(G) \leq \chi(\overline{G})$, consider a matrix X feasible for (53) and a partition $V = Q_1 \cup \ldots \cup Q_k$ into $k := \chi(\overline{G})$ cliques. Then,

$$0 \leq \sum_{h=1}^{k} (k\chi^{Q_h} - e)^T X (k\chi^{Q_h} - e) = k^2 \mathrm{Tr}(X) - k e^T X e = k^2 - k e^T X e,$$

which implies $e^T X e \leq k$ and thus $\vartheta(G) \leq \chi(\overline{G})$.

Several equivalent definitions are known for $\vartheta(G)$ that we recall below. (See [80] or [111] for a detailed treatment.) The dual semidefinite program of (53) reads:

$$\min(t \ \mid tI + \sum_{ij \in E} \lambda_{ij} E_{ij} - J \succeq 0), \tag{55}$$

where $J := ee^T$ is the all ones matrix and E_{ij} is the elementary matrix with all zero entries except 1 at positions (i,j) and (j,i). As the program (53) has a strictly feasible solution (e.g., $X = \frac{1}{n}I$), there is no duality gap and the optimum value of (55) is equal to the theta number $\vartheta(G)$. Setting $Y := J - \sum_{ij \in E} \lambda_{ij} E_{ij}$, $Z := tI - Y$ and $U := \frac{1}{t-1} Z$ in (55), we obtain the following reformulations for $\vartheta(G)$:

$$\begin{array}{rll} \vartheta(G) = & \min & \lambda_{\max}(Y) \\ & \text{s.t.} & Y_{ij} = 1 \ (i = j \ \text{ or } ij \in \overline{E}) \\ & & Y \text{ symmetric matrix,} \end{array} \tag{56}$$

$$\begin{array}{rlllllll} \vartheta(G) = & \min & t & & = & \min & t & \\ & \text{s.t.} & Z_{ii} = t-1 & (i \in V) & & \text{s.t.} & U_{ii} = 1 & (i \in V) \\ & & Z_{ij} = -1 & (ij \in \overline{E}) & & & U_{ij} = -\frac{1}{t-1} & (ij \in \overline{E}) \\ & & Z \succeq 0 & & & & U \succeq 0,\ t \geq 2. & \end{array} \tag{57}$$

The formulation (57) will be used later in Section 6 for the colouring and max k-cut problems. One can also express $\vartheta(G)$ as the optimum value of the linear objective function $e^T x$ maximized over a convex set forming a relaxation of STAB(G). Namely, let $\mathcal{M}_G$ denote the set of positive semidefinite matrices Y indexed by the set $V \cup \{0\}$ satisfying $y_{ii} = y_{0i}$ for $i \in V$ and $y_{ij} = 0$ for $i \neq j \in V$ adjacent in G, and set

$$\text{TH}(G) := \{x \in \mathbb{R}^V \mid \binom{1}{x} = Ye_0 \text{ for some } Y \in \mathcal{M}_G\}, \tag{58}$$

where $e_0 := (1, 0, \ldots, 0)^T \in \mathbb{R}^{n+1}$. (Same definition as (40).)

Lemma 9. *For any graph G,* $\text{STAB}(G) \subseteq \text{TH}(G) \subseteq \text{QSTAB}(G)$.

Proof. If S is a stable set in G and $x := \chi^S$, then $Y := \binom{1}{x}(1\ x^T) \in \mathcal{M}_G$ and $\binom{1}{x} = Ye_0$; from this follows that $\text{STAB}(G) \subseteq \text{TH}(G)$. Let $x \in \text{TH}(G)$, $Y \in \mathcal{M}_G$ such that $\binom{1}{x} = Ye_0$, and let Q be a clique in G. The principal submatrix Y_Q of Y whose rows and columns are indexed by the set $\{0\} \cup Q$ has the form $\begin{pmatrix} 1 & x^T \\ x & \text{diag}(x) \end{pmatrix}$. As $Y \succeq 0$, we have $Y_Q \succeq 0$, i.e., $\text{diag}(x) - xx^T \succeq 0$ (taking a Schur complement), which implies that $e^T(\text{diag}(x) - xx^T)e = e^Tx(1 - e^Tx) \geq 0$ and thus $e^Tx = \sum_{i \in Q} x_i \leq 1$. This shows the inclusion $\text{TH}(G) \subseteq \text{QSTAB}(G)$.

Theorem 10. $\vartheta(G) = \max(e^Tx \mid x \in \text{TH}(G))$.

Proof. We use the formulation of $\vartheta(G)$ from (53). Let μ_G denote the maximum of e^Tx over TH(G). We first show that $\vartheta(G) \leq \mu_G$. For this, let X be an optimum solution to the program (53). Let $v_1, \ldots, v_n \in \mathbb{R}^n$ such that $x_{ij} = v_i^T v_j$ for all $i, j \in V$; thus $\vartheta(G) = || \sum_{i=1}^n v_i||^2$, $\sum_{i=1}^n (v_i)^2 = \text{Tr}(X) = 1$, and $v_i^T v_j = 0$ if i, j are adjacent in G. Set $P := \{i \in V \mid v_i \neq 0\}$, $u_0 := \frac{1}{\sqrt{\vartheta(G)}} \sum_{i=1}^n v_i$, $u_i := \frac{v_i}{||v_i||}$ for $i \in P$, and let u_i $(i \in V \setminus P)$ be an orthonormal basis of the orthogonal complement of the space spanned by $\{v_i \mid i \in P\}$. Let D denote the diagonal matrix indexed by $\{0\} \cup V$ with diagonal entries $u_0^T u_i$ $(i = 0, 1, \ldots, n)$, let Z denote the Gram matrix of $u_0, u_1, \ldots, u_n$ and set $Y := DZD$, with entries $y_{ij} = (u_i^T u_j)(u_0^T u_i)(u_0^T u_j)$ $(i, j = 0, 1, \ldots, n)$. Then, $Y \in \mathcal{M}_G$ with $y_{00} = 1$. It remains to verify that $\vartheta(G) \leq \sum_{i=1}^n y_{0i}$. By the definition of u_0, we find

$$\vartheta(G) = (\sum_{i=1}^n u_0^T v_i)^2 = (\sum_{i \in P} u_0^T v_i)^2 = (\sum_{i \in P} u_0^T u_i \|v_i\|)^2 \leq (\sum_{i \in P} \|v_i\|^2)(\sum_{i \in P} (u_0^T u_i)^2) = \sum_{i=1}^n y_{0i},$$

where the inequality follows using the Cauchy-Schwartz inequality. We now show the converse inequality $\mu_G \leq \vartheta(G)$. For this, let $x \in \text{TH}(G)$ be optimum for the program defining μ_G, let $Y \in \mathcal{M}_G$ such that $\binom{1}{x} = Ye_0$, and $v_0, v_1, \ldots, v_n \in \mathbb{R}^{n+1}$ such that $y_{ij} = v_i^T v_j$ for all $i, j = 0, 1, \ldots, n$. It suffices to construct X feasible for (53) satisfying $\sum_{i,j=1}^n x_{ij} \geq \mu_G$. Define the $n \times n$ matrix X with entries $x_{ij} := \frac{1}{\mu_G} v_i^T v_j$ $(i, j = 1, \ldots, n)$; then X is feasible for (53). Moreover, $\mu_G = \sum_{i=1}^n y_{0i} = \sum_{i=1}^n v_0^T v_i = v_0^T(\sum_{i=1}^n v_i)$ is less than or equal to $\| \sum_{i=1}^n v_i\|$ (by the Cauchy-Schwartz inequality, since $\|v_0\| = 1$).

As $\sum_{i,j=1}^n x_{ij} = \frac{1}{\mu_G}(\sum_{i=1}^n v_i)^2$, we find that $\mu_G \leq \sum_{i,j=1}^n x_{ij}$. ∎

An *orthonormal representation* of G is a set of unit vectors $u_1, \ldots, u_n \in \mathbb{R}^N$ $(N \geq 1)$ satisfying $u_i^T u_j = 0$ for all $ij \in \overline{E}$.

Theorem 11. *$\vartheta(G) = \max_{d,v_i} \sum_{i \in V}(d^T v_i)^2$, where the maximum is taken over all unit vectors $d \in \mathbb{R}^N$ and all orthonormal representations $v_1, \ldots, v_n \in \mathbb{R}^N$ of $\overline{G}$.*

PROOF. Let $\vartheta(G) = e^T X e$, where X is an optimum solution to the program (53) and let $b_1, \ldots, b_n$ be vectors such that $X_{ij} = b_i^T b_j$ for $i, j \in V$. Set $d := \frac{\sum_{i \in V} b_i}{\| \sum_{i \in V} b_i \|}$, $P := \{i \in V \mid b_i \neq 0\}$ and $v_i := \frac{b_i}{\|b_i\|}$ for $i \in P$. Let v_i $(i \in V \setminus P)$ be an orthonormal basis of the orthogonal complement of the space spanned by v_i $(i \in P)$. Then, $v_1, \ldots, v_n$ is an orthonormal representation of $\overline{G}$. We have:

$$\sqrt{\vartheta(G)} = \| \sum_{i \in P} b_i \| = d^T \left(\sum_{i \in P} b_i \right) = \sum_{i \in P} \|b_i\| v_i^T d \leq \sqrt{\sum_{i \in P} \|b_i\|^2} \cdot \sqrt{\sum_{i \in P} (v_i^T d)^2} \leq \sqrt{\sum_{i \in V} (v_i^T d)^2}$$

(using the Cauchy-Schwartz inequality and $\text{Tr}(X) = 1$). This implies that $\vartheta(G) \leq \sum_{i \in V}(d^T v_i)^2$.

Conversely, let d be a unit vector and let $v_1, \ldots, v_n$ be an orthonormal representation of $\overline{G}$. Let Y denote the Gram matrix of the vectors $d, (d^T v_1)v_1, \ldots, (d^T v_n)v_n$. Then, $Y \in \mathcal{M}_G$. Therefore, $((d^T v_1)^2, \ldots, (d^T v_n)^2)^T \in \text{TH}(G)$ which implies that $\sum_{i \in V}(d^T v_i)^2 \leq \vartheta(G)$. ∎

Let A_G denote the convex hull of all vectors $((d^T v_1)^2, \ldots, (d^T v_n)^2)^T$ where d is a unit vector and $v_1, \ldots, v_n$ is an orthonormal representation of $\overline{G}$, let B_G denote the set of $x \in \mathbb{R}_+^V$ satisfying the *orthonormal representation constraints*:

$$\sum_{i \in V} (c^T u_i)^2 x_i \leq 1 \tag{59}$$

for all unit vectors c and all orthonormal representations $u_1, \ldots, u_n$ of G, and let C_G denote the set of $x \in \mathbb{R}_+^V$ satisfying

$$\sum_{i \in V} x_i \leq \min_{c,u_i} \max_{i \in V} \frac{1}{(c^T u_i)^2}$$

where the minimum is taken over all unit vectors c and all orthonormal representations $u_1, \ldots, u_n$ of G.

Lemma 12. *$A_G \subseteq \text{TH}(G) \subseteq B_G \subseteq C_G$.*

PROOF. The inclusion $A_G \subseteq \text{TH}(G)$ follows from the second part of the proof of Theorem 11 and the inclusion $B_G \subseteq C_G$ is easy to verify. Let $x \in \text{TH}(G)$ and let $z := ((c^T u_1)^2, \ldots, (c^T u_n)^2)^T$ where c is a unit vector and $u_1, \ldots, u_n$ is an orthonormal representation of G; we show that $x^T z \leq 1$. By the above, $z \in A_{\overline{G}} \subseteq \text{TH}(\overline{G})$. Let $Y \in \mathcal{M}_G$ and $Z \in \mathcal{M}_{\overline{G}}$ such that $\binom{1}{x} = Y e_0$ and $\binom{1}{z} = Z e_0$. Denote by Y' the matrix obtained from Y by changing the signs on its first row and column. Then, $\langle Y', Z \rangle = 1 - 2\sum_{i \in V} y_{0i} z_{0i} + \sum_{i \in V} y_{ii} z_{ii} = 1 - \sum_{i \in V} x_i z_i \geq 0$ (since $Y', Z \succeq 0$) and thus $x^T z \leq 1$. This shows the inclusion $\text{TH}(G) \subseteq B_G$. ∎

Theorem 13. $\vartheta(G) = \min_{c,u_i} \max_{i\in V} \frac{1}{(c^T u_i)^2}$, *where the minimum is taken over all unit vectors* c *and all orthonormal representations* $u_1, \dots, u_n$ *of* G.

PROOF. The inequality $\vartheta(G) \leq \min \dots$ follows from the inclusion $\mathrm{TH}(G) \subseteq C_G$. For the reverse inequality, we use the definition of $\vartheta(G)$ from (56). Let Y be a symmetric matrix with $Y_{ii} = 1$ $(i \in V)$ and $Y_{ij} = 1$ $(ij \in \overline{E})$ and $\vartheta(G) = \lambda_{\max}(Y)$. As $\vartheta(G)I - Y \succeq 0$, there exist vectors $b_1, \dots, b_n$ such that $b_i^2 = \vartheta(G) - 1$ $(i \in V)$ and $b_i^T b_j = -1$ $(ij \in \overline{E})$. Let c be a unit vector orthogonal to all b_i (which exists since $\vartheta(G)I - Y$ is singular) and set $u_i := \frac{c+b_i}{\sqrt{\vartheta(G)}}$ $(i \in V)$. Then, $u_1, \dots, u_n$ is an orthonormal representation of G and $\vartheta(G) = \frac{1}{(c^T u_i)^2}$ for all i. ∎

Theorems 11 and 13 and Lemma 12 show that one obtains the same optimum value when optimizing the linear objective function $e^T x$ over $\mathrm{TH}(G)$ or over any of the sets A_G, B_G or C_G. In fact, the same remains true for an arbitrary linear objective function $w^T x$ where $w \in \mathbb{R}_+^V$, as the above extends easily to the weighted case. Therefore,

$$\mathrm{TH}(G) = A_G = B_G = C_G.$$

Moreover, $\mathrm{TH}(\overline{G})$ is the *antiblocker* of $\mathrm{TH}(G)$; that is, $\mathrm{TH}(\overline{G}) = \{z \in \mathbb{R}_+^V \mid x^T z \leq 1 \ \forall x \in \mathrm{TH}(G)\}$. One can show that the only orthonormal representation inequalities (59) defining facets of $\mathrm{TH}(G)$ are the clique inequalities. From this follows:

$$\mathrm{TH}(G) \text{ is a polytope } \Longleftrightarrow G \text{ is perfect } \Longleftrightarrow \mathrm{TH}(G) = \mathrm{QSTAB}(G) \Longleftrightarrow \mathrm{TH}(G) = \mathrm{STAB}(G).$$

We refer to ([172], chapter 12) for a detailed exposition on the theta body $\mathrm{TH}(G)$.

4.3 Sharpening the theta function

The number $\vartheta'(G)$. McEliece, Rodemich, Rumsey [142] and Schrijver [174] introduce the parameter $\vartheta'(G)$ as

$$\begin{array}{rll} \vartheta'(G) := & \max & e^T X e \\ & \text{s.t.} & \mathrm{Tr}(X) = 1 \\ & & X_{ij} = 0 \ (i \neq j, \ ij \in E) \\ & & X \succeq 0, \ X \geq 0. \end{array} \tag{60}$$

Comparing with (53), it follows that

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

As was done for $\vartheta(G)$ one can prove the following equivalent formulations for $\vartheta'(G)$:

$$\begin{array}{rll} \vartheta'(G) = & \min & \lambda_{\max}(Y) \\ & \text{s.t.} & Y_{ij} \geq 1 \ (i = j \ \text{ or } ij \in \overline{E}) \\ & & Y \text{ symmetric matrix;} \end{array} \tag{61}$$

$$\begin{array}{rllllll} \vartheta'(G) = & \min & t & & = \ \min & t & \\ & \text{s.t.} & Z_{ii} = t - 1 & (i \in V) & \text{s.t.} & U_{ii} = 1 & (i \in V) \\ & & Z_{ij} \leq -1 & (ij \in \overline{E}) & & U_{ij} \leq -\frac{1}{t-1} & (ij \in \overline{E}) \\ & & Z \succeq 0 & & & U \succeq 0, \ t \geq 2; & \end{array} \tag{62}$$

and $\vartheta'(G) = \max(e^T x \mid \binom{1}{x} = Ye_0$ for some nonnegative matrix $Y \in \mathcal{M}_G\}$. The inequality $\vartheta'(G) \leq \vartheta(G)$ is strict, for instance, for the graph with node set $\{0,1\}^6$ where two nodes are adjacent if their Hamming distance is at most 3 (then, $\vartheta(G) = \frac{16}{3}$ and $\vartheta'(G) = \alpha(G) = 4$).

The number $\vartheta^+(G)$. In a similar vein, Meurdesoif [143] introduces the parameter $\vartheta^+(G)$ which provides a sharper lower bound for the chromatic number of $\overline{G}$:

$$\begin{array}{rll} \vartheta^+(G) := & \min & t \\ & \text{s.t.} & Y_{ii} = t-1 \quad (i \in V) \\ & & Y_{ij} = -1 \quad\;\; (ij \in \overline{E}) \\ & & Y_{ij} \geq -1 \quad\;\; (ij \in E) \\ & & Y \succeq 0. \end{array} \tag{63}$$

Then,

$$\vartheta(G) \leq \vartheta^+(G) \leq \chi(\overline{G}).$$

A graph achieving strict inequality $\vartheta(G) < \vartheta^+(G)$ is given in [143]. The inequality $\vartheta(G) \leq \vartheta^+(G)$ is obvious using (57) and the inequality $\vartheta^+(G) \leq \chi(\overline{G})$ follows from the following formulation of the chromatic number as a bivalent SDP program:

$$\begin{array}{rll} \chi(\overline{G}) = & \min & t \\ & \text{s.t.} & Y_{ii} = t-1 \qquad\qquad (i \in V) \\ & & Y_{ij} = -1 \qquad\qquad\; (ij \in \overline{E}) \\ & & Y_{ij} \in \{-1, t-1\} \quad (ij \in E) \\ & & Y \succeq 0. \end{array} \tag{64}$$

Indeed, let μ_G denote the minimum value of the program (64). If $V = S_1 \cup \ldots \cup S_k$ is a partition into stable sets and if Y is the matrix with entries -1 except $Y_{ij} = k-1$ for $i, j \in S_h$, $h = 1, \ldots, k$, then (Y, k) is feasible for (64) which shows that $\mu_G \leq \chi(\overline{G})$. Conversely, let (t, Y) be an optimum solution for (64), let $v_1, \ldots, v_n \in \mathbb{R}^N$ such that $Y_{ij} = v_i^T v_j$ $(i, j \in V)$, and let $v_{i_1}, \ldots, v_{i_k}$ denote the distinct vectors among $v_1, \ldots, v_n$. For $h \in \{1, \ldots, k\}$, the set $S_h := \{i \in \{1, \ldots, n\} \mid v_i = v_{i_h}\}$ is a clique in G (since $v_i^T v_j = t-1$ if and only if $v_i = v_j$) and thus $k \geq \chi(\overline{G})$. From $0 \leq \| \sum_{h=1}^k v_{i_h} \|^2 = k(t-1) - k(k-1)$ follows that $t \geq k$ and thus $\mu_G \geq \chi(\overline{G})$. Thus equality $\mu_G = \chi(\overline{G})$ holds.

Bounding the Shannon capacity. The theta number $\vartheta(G)$ was introduced by Lovász [134] in connection with a problem of Shannon in coding theory. The strong product $G \cdot H$ of two graphs G and H has node set $V(G) \times V(H)$ with two distinct nodes (u, v) and (u', v') being adjacent if u, u' are equal or adjacent in G and v, v' are equal or adjacent in H. Then G^k is the strong product of k copies of G. The Shannon capacity of G is defined by

$$\Theta(G) := \sup_{k \geq 1} \sqrt[k]{\alpha(G^k)}.$$

As $\alpha(G^k) \geq (\alpha(G))^k$ and $\vartheta(G^k) \leq (\vartheta(G))^k$, one finds

$$\alpha(G) \leq \Theta(G) \leq \vartheta(G).$$

Using these inequalities, Lovász [134] could show that the Shannon capacity of C_5 is $\sqrt{5}$ (as $\alpha(C_5^2) = 5$ and $\theta(C_5) = \sqrt{5}$). For $n \geq 7$ odd, $\vartheta(C_n) = \frac{n \cos(\frac{\pi}{n})}{1 + \cos(\frac{\pi}{n})}$, but the value of $\Theta(C_n)$ is not known.

The theta number versus Delsarte's bound. Let G be a graph whose adjacency matrix can be written as $\sum_{i\in M} A_i$, where $M \subseteq \{1,\dots,N\}$ and $A_0, A_1, \dots, A_N$ are 0/1 symmetric matrices forming an *association scheme*; that is, $A_0 = I$, $\sum_{i=0}^N A_i = J$, there exist scalars p_{ij}^k $(i,j,k = 1,\dots,N)$ such that $A_iA_j = A_jA_i = \sum_{k=0}^N p_{ij}^k A_k$. As the matrices $A_0,\dots,A_N$ commute, they have a common basis of eigenvectors and therefore positive semidefiniteness of a matrix $X := \sum_{i=0}^N x_iA_i$ can be expressed by a linear system of inequalities in $x_1,\dots,x_N$. Therefore, one finds that the theta numbers $\vartheta(G)$, $\vartheta'(G)$ can be computed by solving a linear programming problem. Based on this, Schrijver [174] shows that $\vartheta'(G)$ coincides with a linear programming bound introduced earlier by Delsarte [56].

These ideas have been extended to general semidefinite programs by Goemans and Rendl [74].

4.4 Colouring and finding maximum stable sets in perfect graphs

The stability number $\alpha(G)$ and the chromatic number $\chi(G)$ of a perfect graph G can be computed in polynomial time. (Indeed, it suffices to compute an approximated value of $\vartheta(G)$ with precision $< 1/2$ in order to determine $\alpha(G) = \chi(\overline{G}) = \vartheta(G)$.) We now mention how to find in polynomial time a stable set of size $\alpha(G)$ and a $\chi(G)$-colouring in a perfect graph. The weighted versions of these problems can also be solved in polynomial time (cf. [80] for details).

Finding a maximum cardinality stable set in a perfect graph. Let $G = (V,E)$ be a perfect graph and let $v_1,\dots,v_n$ be an ordering of its nodes. We construct a sequence of graphs $G_0 := G \supseteq G_1 \supset \dots \supseteq G_i \supseteq G_{i+1} \supseteq \dots \supseteq G_n$ in the following manner: For each $i \geq 1$, compute $\alpha(G_{i-1}\backslash v_i)$; if $\alpha(G_{i-1}\backslash v_i) = \alpha(G)$, then set $G_i := G_{i-1}\backslash v_i$, otherwise set $G_i := G_{i-1}$. Then, $\alpha(G_i) = \alpha(G)$ for all i and G_n is a stable set, thus providing a maximum stable set in G. Therefore, a maximum stable set in a perfect graph G can be found by applying n times an algorithm for computing the theta function.

Finding a minimum colouring in a perfect graph. We follow the presentation of Schrijver [176]. Let $G = (V,E)$ be a perfect graph. A crucial observation is that it suffices to find a stable set S which intersects all the maximum cardinality cliques of G. Indeed, if such S is found, then one can recursively colour $G\backslash S$ with $\omega(G\backslash S) = \omega(S) - 1$ colours and thus G with $\omega(G) = \chi(G)$ colours. For $t \geq 1$, we grow iteratively a list $Q_1,\dots,Q_t$ of maximum cardinality cliques. Suppose $Q_1,\dots,Q_t$ have been found. We begin with finding a stable set S meeting each of $Q_1,\dots,Q_t$. For this, setting $w := \sum_{i=1}^t \chi^{Q_i}$, it suffices to find a maximum weight stable set S. (This can be done by applying the above maximum cardinality stable set algorithm to the graph G' obtained from G by replacing every node i by a set W_i of w_i nonadjacent nodes, making two nodes $u \in W_i$, $v \in W_j$ adjacent in G' if the nodes i,j are adjacent in G.) Then S has weight t which means that S meets each of $Q_1,\dots,Q_t$. Now, if $\omega(G\backslash S) < \omega(G)$, then S meets *all* the maximum cardinality cliques in G and we are done. Otherwise, we find a clique Q_{t+1} in $G\backslash S$ of size $\omega(G)$ and add it to our list.

The algorithm has a polynomial running time since the number of iterations is bounded by $|V|$. To see it, consider the affine space $L_t := \{x \in \mathbb{R}^V \mid x(Q_i) = 1 \ \forall i = 1,\dots,t\}$. Then, $L_1 \supseteq L_2 \supseteq \dots \supseteq L_t \supseteq L_{t+1} \supseteq \dots$. The dimension of the spaces L_t decreases at each step since $\chi^S \in L_t \setminus L_{t+1}$, where S is the stable set constructed at the t-th iteration as above.

5 Semidefinite Relaxation for the Max-Cut Problem

We present here results dealing with the basic semidefinite relaxation of the cut polytope and its application to desigining good approximation algorithms for the max-cut problem.

Given a graph $G = (V, E)$, the *cut* $\delta(S)$ induced by a vertex set $S \subseteq V$ is the set of edges with exactly one endpoint in S. Given edge weights $w \in \mathbb{Q}^E$, the *max-cut problem* consists of finding a cut $\delta(S)$ whose weight $w(\delta(S)) := \sum_{ij \in \delta(S)} w_{ij}$ is maximum. Let $\mathrm{mc}(G, w)$ denote the maximum weight of a cut in G. A comprehensive survey about the max-cut problem can be found in [163]. The max-cut problem is one of the basic NP-hard problems studied by Karp [103]. Moreover, it cannot be approximated with an arbitrary precision; namely, Håstad [89] shows that for $\rho > \frac{16}{17} = 0.94117$ there is no ρ-approximation algorithm for max-cut if P $\neq$ NP. [A ρ-approximation algorithm is an algorithm that returns in polynomial time a cut whose weight is at least ρ times the maximum weight of a cut; ρ being called the *performance ratio* or *guarantee*.] On the other hand, Goemans and Williamson [77] prove a 0.878-approximation algorithm for max-cut that will be presented in Section 5.3 below.

5.1 The basic linear relaxation

As before, the *cut polytope* $\mathrm{CUT}(G)$ is the polytope in $\mathbb{R}^E$ defined as the convex hull of the vectors $z^S \in \{\pm 1\}^E$ for $S \subseteq V$, where $z^S_{ij} = -1$ if and only if $|S \cap \{i, j\}| = 1$. The weight of the cut $\delta(S)$ can be expressed as

$$\frac{1}{2} \sum_{ij \in E} w_{ij}(1 - z_{ij}). \tag{65}$$

Hence the max-cut problem is the problem of optimizing the linear objective function (65) over $\mathrm{CUT}(G)$. The *circuit inequalities*:

$$\sum_{ij \in F} x_{ij} - \sum_{ij \in E(C) \setminus F} x_{ij} \geq 2 - |C|, \tag{66}$$

where C is a circuit in G and F is a subset of $E(C)$ with an odd cardinality, are valid for $\mathrm{CUT}(G)$ as they express the fact that a cut and a circuit must have an even intersection. Together with the bounds $-1 \leq x_{ij} \leq 1$ ($ij \in E$) they define the *metric polytope* $\mathrm{MET}(G)$. Thus $\mathrm{CUT}(G) \subseteq \mathrm{MET}(G)$; moreover, the only ± 1 vectors in $\mathrm{MET}(G)$ are the cut vectors z^S ($S \subseteq V$). An inequality (66) defines a facet of $\mathrm{CUT}(G)$ if and only if C is a chordless circuit in G while an inequality $\pm x_{ij} \leq 1$ is facet defining if and only if ij does not belong to a triangle [23]. Hence the metric polytope $\mathrm{MET}(K_n)$ is defined by the $4\binom{n}{3}$ *triangle inequalities*:

$$x_{ij} + x_{ik} + x_{jk} \geq -1, \;\; x_{ij} - x_{ik} - x_{jk} \geq -1 \tag{67}$$

for all triples $i, j, k \in \{1, \ldots, n\}$. Therefore, one can optimize any linear objective function over $\mathrm{MET}(K_n)$ in polynomial time. The same holds for $\mathrm{MET}(G)$, since $\mathrm{MET}(G)$ is equal to the projection of $\mathrm{MET}(K_n)$ on the subspace $\mathbb{R}^E$ indexed by the edge set of G [20]. The inclusion $\mathrm{CUT}(G) \subseteq \mathrm{MET}(G)$ holds at equality if and only if G has no K_5-minor [23]. Therefore, the max-cut problem can be solved in polynomial time for the graphs with no K_5-minor (including the planar graphs).

Let $\mathrm{met}(G, w)$ denote the optimum value of (65) maximized over $z \in \mathrm{MET}(G)$. When all edge weights are equal to 1, we also use the notation $\mathrm{met}(G)$ in place of $\mathrm{met}(G, w)$ (and analogously $\mathrm{mc}(G)$ in place of $\mathrm{mc}(G, w)$). How well does the polyhedral bound $\mathrm{met}(G, w)$ approximate the max-cut value

$\mathrm{mc}(G,w)$? In order to compare the two bounds, we assume that all edge weights are nonnegative. Then,

$$\mathrm{met}(G,w) \leq w(E) = \sum_{ij\in E} w_{ij} \text{ and } \mathrm{mc}(G,w) \geq \frac{1}{2}w(E).$$

(To see the latter inequality, consider an optimum cut $\delta(S)$ and the associated partition $(S, V \setminus S)$. Then, for every node $i \in V$, the sum of the weights of the edges connecting i to the opposite class of the partition is greater than or equal to the sum of the weights of the edges connecting i to nodes in the same class, since otherwise moving i to the other class would produce a heavier cut.) Therefore,

$$\frac{\mathrm{mc}(G,w)}{\mathrm{met}(G,w)} \geq \frac{1}{2}.$$

In fact, the ratio $\frac{\mathrm{mc}(G,w)}{\mathrm{met}(G,w)}$ tends to $\frac{1}{2}$ for certain classes of graphs (cf. [160], [162]) which shows that in the worst case the metric polytope does not provide a better approximation than the trivial relaxation of $\mathrm{CUT}(G)$ by the cube $[-1,1]^E$.

The polytope

$$Q(G) := \{x \in [-1,1]^E \mid \sum_{ij\in E(C)} x_{ij} \geq 2 - |C| \text{ for all odd circuits } C \text{ in } G\}$$

contains the metric polytope $\mathrm{MET}(G)$ and its ±1-vectors correspond to the bipartite subgraphs of G. Therefore, the max-cut problem for nonnegative weights can be reformulated as the problem of maximizing (65) over the ±1-vectors in $Q(G)$. A graph G is said to be *weakly bipartite* when all the vertices of $Q(G)$ are ±1-valued. It is shown in [81] that one can optimize in polynomial time a linear objective function over $Q(G)$. Therefore, the max-cut problem can be solved in polynomial time for weakly bipartite graphs with nonnegative edge weights. Guenin [83] characterized the weakly bipartite graphs as those graphs *containing no odd* K_5-minor (they include the graphs with no K_5-minor, the graphs having two nodes covering all odd circuits, etc.), settling a conjecture posed by Seymour [177]. Poljak [160] shows that, for nonnegative edge weights, one obtains in fact the same optimum value when optimizing (65) over $\mathrm{MET}(G)$ or over $Q(G)$.

5.2 The basic semidefinite relaxation

The max-cut problem can be reformulated as the following integer quadratic program:

$$\begin{array}{rll} \mathrm{mc}(G,w) = & \max & \frac{1}{2}\sum_{ij\in E} w_{ij}(1 - x_i x_j) \\ & \text{s.t.} & x_1,\ldots,x_n \in \{\pm 1\}. \end{array} \tag{68}$$

For $x \in \{\pm1\}^n$, the matrix $X := xx^T$ is positive semidefinite with all diagonal elements equal to one. Thus relaxing the rank one condition on X, we obtain the following *semidefinite relaxation* for max-cut:

$$\begin{array}{rll} \mathrm{sdp}(G,w) := & \max & \frac{1}{2}\sum_{ij\in E} w_{ij}(1 - x_{ij}) \\ & \text{s.t.} & x_{ii} = 1 \ \forall i \in \{1,\ldots,n\} \\ & & X = (x_{ij}) \succeq 0. \end{array} \tag{69}$$

The set

$$\mathcal{E}_n := \{X = (x_{ij})_{i,j=1}^n \mid X \succeq 0 \text{ and } x_{ii} = 1 \ \forall i \in \{1,\ldots,n\}\} \tag{70}$$

is the basic semidefinite relaxation of the cut polytope $\mathrm{CUT}(K_n)$. More precisely,

$$x \in \mathrm{CUT}(K_n) \Longrightarrow \mathrm{mat}(x) \in \mathcal{E}_n \tag{71}$$

where $\mathrm{mat}(x)$ is the $n \times n$ symmetric matrix with ones on its main diagonal and x_{ij} as off-diagonal entries.

The quantity $\mathrm{sdp}(G,w)$ can be computed in polynomial time (with an arbitrary precision). The objective function in (69) is equal to $\frac{1}{4}\langle L_w, X\rangle$, where $L_w = (l_{ij})$ is the *Laplacian* matrix defined by $l_{ii} := w(\delta(i))$ and $l_{ij} := -w_{ij}$ for $i \neq j$ (assigning weight 0 to non edges). Hence, the dual of the semidefinite program (69) is

$$\frac{1}{4}\min\{\sum_{i=1}^{n} y_i \mid \mathrm{diag}(y) - L_w \succeq 0\} \tag{72}$$

and there is no duality gap (since I is a stricty feasible solution to (69)). Set $s = \frac{1}{n}y^Te$ and $u = se - y$; then $u^Te = 0$ and $\mathrm{diag}(y) - L_w = sI - \mathrm{diag}(u) - L_w \succeq 0$ if and only if $\lambda_{\max}(L_w + \mathrm{diag}(u)) \leq s$. Therefore, (72) can be rewritten as the following eigenvalue optimization problem:

$$\frac{n}{4}\min\{\lambda_{\max}(L_w + \mathrm{diag}(u)) \mid \sum_{i=1}^{n} u_i = 0\}; \tag{73}$$

this eigenvalue upper bound for max-cut had been introduced and studied earlier by Delorme and Poljak [53, 54]. One can also verify directly that (73) is an upper bound for max-cut. Indeed, for $x \in \{\pm 1\}^n$ and $u \in \mathbb{R}^n$ with $\sum_i u_i = 0$, one has:

$$w(\delta(S)) = \frac{1}{4}x^TL_wx = \frac{1}{4}x^T(L_w + \mathrm{diag}(u))x = \frac{n}{4}\frac{x^T(L_w + \mathrm{diag}(u))x}{x^Tx}$$

which is less than or equal to $\frac{n}{4}\lambda_{\max}(L_w + \mathrm{diag}(u))$ by the Rayleigh principle. The program (73) can be shown to have a unique minimizer u (when $w \neq 0$); this minimizer u is equal to the null vector, for instance, when G is vertex transitive, in which case the computation of the semidefinite bound amounts to an eigenvalue computation [53]. Based on this, one can compute the semidefinite bound for unweighted circuits. Namely, $\mathrm{mc}(C_{2k}) = \mathrm{sdp}(C_{2k}) = 2k$ and $\mathrm{mc}(C_{2k+1}) = 2k$ while $\mathrm{sdp}(C_{2k+1}) = \frac{2k+1}{4}(2 + 2\cos(\frac{\pi}{2k+1}))$. Hence, $\frac{\mathrm{mc}(C_5)}{\mathrm{sdp}(C_5)} = \frac{32}{25+5\sqrt{5}} \sim 0.88445$; the same ratio is obtained for some other circulant graphs [144].

Much research has been done for evaluating the *integrality ratio* $\frac{\mathrm{mc}(G,w)}{\mathrm{sdp}(G,w)}$ and comparing the polyhedral and semidefinite bounds. Poljak [160] proved the following inequality relating the two bounds:

$$\frac{\mathrm{met}(G,w)}{\mathrm{sdp}(G,w)} \geq \frac{32}{25 + 5\sqrt{5}} \quad \text{for any graph } G \text{ and } w \geq 0. \tag{74}$$

Therefore, the inequality

$$\frac{\mathrm{mc}(G,w)}{\mathrm{sdp}(G,w)} \geq \frac{32}{25 + 5\sqrt{5}} \tag{75}$$

holds for any weakly bipartite graph (G,w) with $w \geq 0$. The bound (75) remains valid for unweighted line graphs and the better bound $\frac{8}{9}$ was proved for the complete graph K_n with edge weights $w_{ij} := b_ib_j$ (given $b_1, \ldots, b_n \in \mathbb{R}_+$) or for Paley graphs [53]. Moreover, the integrality ratio is asymptotically equal to 1 for the random graphs $G_{n,p}$ (p denoting the edge probability) [53].

Goemans and Williamson [77] proved the following bound for the integrality ratio:

$$\frac{\mathrm{mc}(G,w)}{\mathrm{sdp}(G,w)} \geq \alpha_0 \quad \text{for any graph } G \text{ and } w \geq 0, \tag{76}$$

where $0.87856 < \alpha_0 < 0.87857$ and α_0 is defined by

$$\alpha_0 := \min_{0<\theta\leq\pi} \frac{2}{\pi}\frac{\theta}{1-\cos\theta}. \tag{77}$$

Moreover, they present a randomized algorithm producing a cut whose expected weight is at least $\alpha_0 \cdot \mathrm{sdp}(G,w)$; their result will be described in the next subsection.

Until recently, no example was known of a graph having a worst integrality ratio than C_5 and it had been conjectured by Delorme and Poljak [53] that $\frac{32}{25+5\sqrt{5}}$ is the worst possible value for the integrality ratio. Recently, Feige and Schechtman [66] disproved this conjecture and proved that the worst case value for the integrality ratio $\frac{\mathrm{mc}(G,w)}{\mathrm{sdp}(G,w)}$ is *equal* to the Goemans-Williamson quantity α_0; we will come back to this result later in this section.

5.3 The Goemans-Williamson randomized approximation algorithm for max-cut

The randomized approximation algorithm of Goemans and Williamson [77] for max-cut goes as follows; its analysis will need the assumption that the edge weights are nonnegative.

1. *The semidefinite optimization phase:* Solve the semidefinite program (69). Let $X = (x_{ij})$ be an optimum solution and let $v_1, \ldots, v_n \in \mathbb{R}^d$ (for some $d \leq n$) such that $x_{ij} = v_i^T v_j$ for all $i, j \in \{1, \ldots, n\}$.

2. *The random hyperplane rounding phase:* Generate a random unit vector r and set $S := \{i \mid v_i^T r \geq 0\}$. Then, $\delta(S)$ is the randomized cut returned by the algorithm.

The hyperplane H_r with normal r cuts the space into two half-spaces and an edge ij belongs to the cut $\delta(S)$ if and only if the vectors v_i and v_j do not belong to the same half-space. Hence the probability that an edge ij belongs to $\delta(S)$ is equal to $\frac{\arccos(v_i^T v_j)}{\pi}$ and the expected weight $E(w(S))$ of the cut $\delta(S)$ is equal to

$$E(w(S)) = \sum_{ij\in E} w_{ij} \frac{\arccos(v_i^T v_j)}{\pi} = \sum_{ij\in E} w_{ij} \frac{1 - v_i^T v_j}{2} \cdot \frac{2}{\pi} \frac{\arccos(v_i^T v_j)}{1 - v_i^T v_j} \geq \alpha_0 \cdot \mathrm{sdp}(G,w).$$

The last inequality holds if we assume that $w \geq 0$. As $E(w(S)) \leq \mathrm{mc}(G,w)$, we find

$$\frac{\mathrm{mc}(G,w)}{\mathrm{sdp}(G,w)} \geq \frac{E(w(S))}{\mathrm{sdp}(G,w)} \geq \alpha_0 > 0.87856. \tag{78}$$

As a biproduct of the analysis, we obtain the following trigonometric reformulation for max-cut with $w \geq 0$:

$$\begin{array}{rll} \mathrm{mc}(G,w) = & \max & \sum_{ij\in E} w_{ij} \frac{\arccos(v_i^T v_j)}{\pi} \\ & \text{s.t.} & v_1, \ldots, v_n \text{ unit vectors in } \mathbb{R}^n. \end{array} \tag{79}$$

Mahajan and Ramesh [139] have shown that the above randomized algorithm can be derandomized, therefore giving a deterministic α_0-approximation algorithm for max-cut. Let us stress that until then the best known approximation algorithm was the simple random partition algorithm (which assigns a node to either side of the partition independently with probability $\frac{1}{2}$) with a performance ratio of $\frac{1}{2}$.

As mentioned above, the integrality ratio $\frac{\mathrm{mc}(G,w)}{\mathrm{sdp}(G,w)}$ is equal to α_0 in the worst case. More precisely, Feige and Schechtman [66] show that for every $\epsilon > 0$ there exists a graph G (unweighted) for which the ratio is at most $\alpha_0 + \epsilon$. The basic idea of their construction is as follows. Let θ_0 denote the angle where the minimum in the definition of $\alpha_0 = \min_{0<\theta\leq\pi} \frac{2}{\pi}\frac{\theta}{1-\cos\theta}$ is attained; $\theta_0 \sim 2,331122$ is the nonzero root of $\cos\theta + \theta\sin\theta = 1$. Let $[\theta_1, \theta_2]$ be the largest interval containing θ_0 satisfying $\theta \in [\theta_1, \theta_2] \Longrightarrow \frac{2}{\pi}\frac{\theta}{1-\cos\theta} \leq \alpha_0 + \epsilon$. Distribute n points $v_1, \ldots, v_n$ uniformly on the unit sphere S^{d-1} in $\mathbb{R}^d$ and let G be the graph on n nodes where there is an edge ij if and only if the angle between v_i and v_j belongs to $[\theta_1, \theta_2]$. Applying the random hyperplane rounding phase to the vectors $v_1, \ldots, v_n$, the above analysis shows that the expected weight of the returned cut satisfies

$$\frac{E(w(S))}{\mathrm{sdp}(G)} \leq \alpha_0 + \epsilon.$$

The crucial part of the proof consists then of showing that for some suitable choice of the dimension d and of the distribution of the n points on the sphere S^{d-1} the expected weight $E(w(S))$ is not far from the max-cut value $mc(G)$.

Nesterov [149] shows the weaker bound:

$$\frac{E(w(S))}{\mathrm{sdp}(G,w)} \geq \frac{2}{\pi} \sim 0.63661 \tag{80}$$

for the larger class of weight functions w satisfying $L_w \succeq 0$. (Note indeed that $L_w \succeq 0$ if $w \geq 0$.) Hence, the GW rounding technique applies to a larger class of instances at the cost of obtaining a weaker performance ratio. Cf. Section 6.1 for more details.

The above analysis of the GW algorithm shows that its performance guarantee is at least α_0. Karloff [101] shows that it is, in fact, equal to α_0. For this, he constructs a class of graphs G (edge weights are equal to 1) for which the ratio $\frac{E(w(S))}{\mathrm{sdp}(G,w)}$ can be made arbitrarily close to α_0. (The graphs constructed by Feige and Schechtman [66] display the same behaviour; the construction of Karloff has however a simpler proof.) These graphs are the Johnson graphs $J(m, \frac{m}{2}, b)$ for m even, $b \leq \frac{m}{12}$ having the collection of subsets of $\{1, \ldots, m\}$ of cardinality $\frac{m}{2}$ as node set and two nodes being adjacent if their intersection has cardinality b. An additional feature of these graphs is that $\mathrm{mc}(G,w) = \mathrm{sdp}(G,w)$. Hence, one of the problems that the Karloff's example emphasizes is that although the semidefinite program already solves the max-cut problem at optimality, the GW approximation algorithm is not able to recognize this fact and to take advantage of it for producing a better cut. As a matter of fact, recognizing whether $\mathrm{sdp}(G,w) = \mathrm{mc}(G,w)$ for given weights w is an NP-complete problem [54, 126].

Goemans and Williamson [77] show that their algorithm behaves, in fact, better for graphs having $\frac{\mathrm{sdp}(G,w)}{w(E)} \geq \frac{85}{100}$ (and thus for graphs having very large cuts). To express their result, set $h(t) := \frac{1}{\pi}\arccos(1-2t)$, $t_0 := \frac{1-\cos\theta_0}{2} \sim 0.84458$, where $\theta_0 \sim 2,331122$ is the angle at which the minimum in the definition of $\alpha_0 = \min_{0<\theta\leq\pi} \frac{2}{\pi}\frac{\theta}{1-\cos\theta}$ is attained. Then, $\frac{h(t_0)}{t_0} = \alpha_0$ and it follows from the definition of α_0 that $h(t) \geq \alpha_0 t$ for $t \in [0,1]$. Further, set

$$\alpha_{\mathrm{GW}}(t) := \frac{h(t)}{t} \text{ if } t \in [t_0, 1] \text{ and } \alpha_{\mathrm{GW}}(t) := \alpha_0 \text{ if } t \in [0, t_0].$$

One can verify that the function $\tilde{h}(t) := \alpha_{\text{GW}}(t)t$ is convex on $[0,1]$ and $\tilde{h} \leq h$. From this follows that

$$\frac{E(w(S))}{\text{sdp}(G,w)} \geq \alpha_{\text{GW}}(A), \text{ where } A := \frac{\text{sdp}(G,w)}{w(E)}. \tag{81}$$

Indeed, setting $y_{ij} := \frac{1-v_i^T v_j}{2}$, we have:

$$\frac{E(w(S))}{w(E)} = \sum_{ij \in E} \frac{w_{ij}}{w(E)} h(y_{ij}) \geq \sum_{ij \in E} \frac{w_{ij}}{w(E)} \tilde{h}(y_{ij}) \geq \tilde{h}\left(\sum_{ij \in E} \frac{w_{ij}}{w(E)} y_{ij}\right) = \tilde{h}(A) = \alpha_{\text{GW}}(A) \cdot A$$

which implies (81). Therefore, the performance guarantee of the GW algorithm is at least $\alpha_{\text{GW}}(A)$ which is greater than α_0 when $A > t_0$ and tends to 1 as A tends to 1. Extending Karloff's result, Alon and Sudakov [7] construct (unweighted) graphs G for which $\text{mc}(G,w) = \text{sdp}(G,w)$ and $\frac{E(w(S))}{\text{sdp}(G,w)} = \alpha_{\text{GW}}(A)$ for any $A = \frac{\text{sdp}(G,w)}{w(E)} \geq t_0$, which shows that the performance guarantee of the GW algorithm is equal to $\alpha_{\text{GW}}(A)$. For the remaining values of A, $\frac{1}{2} \leq A < t_0$, Alon, Sudakov and Zwick [8] construct graphs satisfying $\text{mc}(G,w) = \text{sdp}(G,w)$ and $\frac{E(w(S))}{\text{sdp}(G,w)} = \alpha_0$ which shows that the analsyis of Goemans and Williamson is also tight in this case.

5.4 How to improve the Goemans-Williamson algorithm?

There are several ways in which one can try to modify the basic algorithm of Goemans and Williamson in order to obtain an approximation algorithm with a better performance ratio.

Adding valid inequalities. Perhaps the most natural idea is to strengthen the basic semidefinite relaxation by adding inequalities valid for the cut polytope. For instance, one can add all triangle inequalities; denote by $\text{sdp}'(G,w)$ the optimum value of the semidefinite program obtained by adding the triangle inequalities to (69). The new integrality ratio $\frac{\text{mc}(G,w)}{\text{sdp}'(G,w)}$ is equal to 1 for graphs with no K_5-minor (thus for C_5). For K_5 (with edge weights 1) it is equal to $\frac{24}{25} = 0.96$. However this is not the worst case; Feige and Schechtman [66] construct graphs for which the new integrality ratio is no better than roughly 0.891.

On the other hand, the example of Karloff shows that the GW randomized approximation algorithm applied to the tighter semidefinite relaxation does not have a better performance guarantee. The same remains true if we would add to the semidefinite relaxation *all* inequalities valid for the cut polytope (because the Karloff's graphs satisfy $\frac{E(w(S))}{\text{sdp}(G,w)} \sim \alpha_0$ while $\text{mc}(G,w) = \text{sdp}(G,w)$!). Therefore, in order to improve the performance guarantee, beside adding some valid inequalities, a new rounding technique will be needed. We now present two ideas along these lines: the first from [64] uses triangle inequalities and adds a 'local search' phase to the GW algorithm, the second from [202] can be seen as a mixing of the hyperplane rounding technique and the basic random algorithm.

Adding valid inequalities and a local search phase. Feige, Karpinski and Langberg [64] have presented an approximation algorithm for max-cut with a better performance guarantee for graphs with a bounded maximum degree Δ (edge weights are assumed to be equal to one). Their algorithm has two new features: triangle inequalities are added to the basic semidefinite relaxation (also some triangle equalities in the case $\Delta = 3$) and an additional 'greedy' phase is added after the GW hyperplane rounding phase.

Given a partition $(S, S \setminus S)$, a vertex v belonging, say, to S, is called *misplaced* if it has more neighbours in S than in $V \setminus S$; then the cut $\delta(S \setminus \{v\})$ has more edges than the cut $\delta(S)$. One of the basic ideas underlying the FKL algorithm is that, if $(S, V \setminus S)$ is the partition produced by the hyperplane rounding phase and if all angles $\arccos(v_i^T v_j)$ are equal to θ_0 (which implies $E(w(S)) = \alpha_0 \cdot \mathrm{sdp}(G,w)$), then there is a positive probability (depending on Δ alone) of finding a misplaced vertex in the partition and, therefore, one can improve the cut.

In the case $\Delta = 3$ the FKL algorithm goes as follows. In the first step one solves the semidefinite program (69) to which have been added all triangle inequalities as well as the triangle equalities $x_{ij} + x_{ik} + x_{jk} = -1$ for all triples (i,j,k) for which $ij, ik \in E$ (such equality is indeed valid for a maximum cut for, if not, the vertex i would be misplaced). Then the hyperplane rounding phase is applied to the optimum matrix X, producing a partition $(S, V \setminus S)$. After that comes an additional *greedy phase*: If the partition $(S, V \setminus S)$ has a misplaced vertex v, move it to the other side of the partition and repeat until no misplaced vertex can be found. If at some step there several misplaced vertices, we move the misplaced vertex v for which the ratio between the number of edges gained in the cut by moving v and the number of triples (i,j,k) with $ij, ik \in E$ and i misplaced destroyed by this action, is maximal.
It is shown in [64] that the expected weight of the final partition returned by the FKL algorithm satisfies

$$E(w(S)) \geq 0.919 \cdot \mathrm{sdp}(G,w). \tag{82}$$

For regular graphs of degree 3, one can show an approximation ratio of 0.924 and, for graphs with maximum degree Δ, a ratio of $\alpha_0 + \frac{1}{2^{33}\Delta^4}$. Note that, when $\Delta \geq 4$, one cannot incorporate the triangle equality $x_{ij} + x_{ik} + x_{jk} = -1$ (with $ij, ik \in E$) as it is no longer valid for maximum cuts.

Recently, Halperin, Livnat and Zwick [85] gave an improved approximation algorithm for max-cut in graphs of maximum degree 3 with performance guarantee 0.9326. Their algorithm has an additional preprocessing phase (which converts the input graph into a cubic graph satisfying some additional property) and performs the greedy phase in a more global manner; moreover, it applies to a more general problem than max-cut.

Mixing the random hyperplane and the basic random rounding techniques. We saw above that the performance guarantee of the GW algorithm is greater than α_0 for graphs with large cuts (with weight at least 85% of the total weight of edges). Zwick [202] presents a modification of the GW algorithm which, on the other hand, has a better performance guarantee for graphs having *no* large cuts.

Note that the simple randomized algorithm, which constructs a partition $(S, V \setminus S)$ by assigning a vertex with probability $\frac{1}{2}$ to either side of the partition, produces a cut with expected weight $\frac{w(E)}{2}$ and thus its performance ratio is

$$\alpha_{\mathrm{rand}}(A) := \frac{1}{2A} \text{ where } A = \frac{\mathrm{sdp}(G,w)}{w(E)}.$$

Note, moreover, that this algorithm is equivalent to applying the hyperplane rounding technique to the standard unit vectors $e_1, \ldots, e_n$, with the identity matrix as Gram matrix. As $\alpha_{\mathrm{rand}}(A) \geq \alpha_{\mathrm{GW}}(A)$ when $\frac{1}{2} \leq A \leq \frac{1}{2\alpha_0} \sim 0.569113$, Zwick's idea is to make a 'mix' of the hyperplane rounding and basic random algorithms. For this, if X is the optimum matrix obtained when solving the basic semidefinite program (69), set

$$X' := (\cos^2 \gamma_A)X + (\sin^2 \gamma_A)I$$

where $\gamma_A \in [0, \pi]$ is suitable chosen. Namely, if $A \geq t_0$ then $\gamma_A := 0$ and if $\frac{1}{2}A \leq t_0$, then solve the following equations for c and t:

$$\frac{\arccos(c(1-2t)) - \arccos c}{t} = \frac{2c}{\sqrt{1-c^2(1-2t)^2}}, \quad \frac{1-\frac{t}{A}}{\sqrt{1-c^2}} = \frac{1-2t}{\sqrt{1-c^2(1-2t)^2}}$$

(there is a unique solution c_A, t_A such that $0 \leq c_A \leq 1$ and $\frac{3}{4} \leq t_A \leq t_0$) and set $\gamma_A := \arccos(\sqrt{c_A})$. Note that γ_A tends to $\frac{\pi}{2}$ as A tends to $\frac{1}{2}$. Then a randomized cut $\delta(S)$ is produced by applying the hyperplane rounding phase to the modified matrix X'. Zwick shows that

$$\frac{E(w(S))}{\mathrm{sdp}(G,w)} \geq \alpha_{\mathrm{rot}}(A) \text{ for any graph } G \text{ and } w \geq 0 \tag{83}$$

where $\alpha_{\mathrm{rot}}(A) := \alpha_{\mathrm{GW}}(A)$ for $A \geq t_0$ and, setting $h_c(t) := \frac{\arccos(c(1-2t))}{\pi}$,

$$\alpha_{\mathrm{rot}}(A) := (\frac{1}{A} - \frac{1}{t_A})h_{c_A}(0) + \frac{1}{t_A}h_{c_A}(t_A)$$

for $\frac{1}{2} \leq A \leq t_0$. The new performance guarantee is at least $\alpha_{\mathrm{rot}}(A)$, which is greater than $\alpha_{\mathrm{rand}}(A)$ and $\alpha_{\mathrm{GW}}(A)$ when $A < t_0$. For instance, $\alpha_{\mathrm{rot}}(A) \geq 0.88$ if $A \leq 0.75$, $\alpha_{\mathrm{rot}}(A) \geq 0.91$ if $A \leq 0.6$. Alon, Sudakov and Zwick [8] show that the analysis is tight; for this they construct graphs having $\mathrm{mc}(G,w) = \mathrm{sdp}(G,w)$ and $\frac{E(w(S))}{\mathrm{sdp}(G,w)} = \alpha_{\mathrm{rot}}(A)$ for any $\frac{1}{2} \leq A \leq t_0$.

Inapproximability results. Summarizing, the best performance guarantee of an approximation algorithm for max-cut known so far is $\alpha_0 \sim 0.87856$. In fact, $\frac{16}{17} \sim 0.94117$ is the best performance guarantee that one can hope for. Indeed, Håstad [89] shows that, for any $\epsilon > 0$, there is no $(\frac{16}{17} + \epsilon)$-approximation algorithm for max-cut if $P \neq NP$. Berman and Karpinski [31] show that it is NP-hard to approximate max-cut in cubic graphs beyond the ratio of 0.997 (while there is an 0.932-approximation algorithm as we saw above).

On the positive side, Arora, Karger and Karpinski [15] show that the max-cut problem has a polynomial time approximation scheme (that is, an $(1-\epsilon)$-approximation algorithm for any $\epsilon > 0$) when restricted to dense graphs, that is, graphs with $O(n^2)$ edges. De la Vega [192] described independently a randomized approximation scheme for max-cut in graphs with minimum degree cn for some constant $c > 0$.

We have seen in Section 3.6 several techniques permitting to construct semidefinite relaxations of the cut polytope refining the basic one. Thus a natural and very interesting question is whether some of them can be used for proving a better integrality ratio (better than the Goemans-Williamson bound α_0) and for designing an approximation algorithm for max-cut with an improved performance ratio. The most natural candidate to consider might be the Lasserre relaxation $Q_1(K_n)$ (defined using (43) and (44)) or its subset, the Anjos-Wolkowicz relaxation F_n (defined using (43)).

6 Applications of Semidefinite Programming and the Rounding Hyperplane Technique to Other Combinatorial Optimization Problems

The method developped by Goemans and Williamson for approximating the max-cut problem has been applied and generalized to a large number of combinatorial optimization problems. Summarizing, their method consists of the following two phases:

1. *The semidefinite optimization phase*, which finds a set of vectors $v_1, \ldots, v_n$ providing a Cholesky factorization of an optimum solution to the SDP program relaxing the original combinatorial problem.

2. *The random hyperplane rounding phase*, which constructs a solution to the original combinatorial problem by looking at the positions of the vectors v_i with respect to some random hyperplane.

The basic method of Goemans and Williamson may have to be modified in order to be applied to some other combinatorial problems. In the first phase, one has to choose an appropriate SDP relaxation of the problem at hand and, in the second phase, one may have to adapt the rouding procedure. For instance, if one wants to approximate graph colouring and max k-cut problems, one should consider more general partitions of the space using more than one random hyperplane. One may also have to add an additionnal phase permitting to modify the returned solution; for instance, to turn the returned cut into a bisection if one wants to approximate the bisection problem. It turns out that the analysis of the extended approximation algorithms is often more complicated than that of the basic GW algorithm; it sometimes needs the evaluation of certain integral formulas that are hard to evaluate numerically.

In this section we present approximation algorithms based on these ideas for the following problems: general quadratic programming problems, maximum bisection and k-cut problems, colouring, stable sets, MAX SAT and maximum directed cut problems.

6.1 Approximating quadratic programming

We consider here the boolean quadratic programming problem:

$$\begin{array}{rll} m^*(A) := & \max & x^T A x \\ & \text{s.t.} & x_1, \ldots, x_n \in \{\pm 1\}^n \end{array} \tag{84}$$

where A is a symmetric matrix of order n, and its natural SDP relaxation:

$$\begin{array}{rll} s^*(A) := & \max & \langle A, X \rangle \\ & \text{s.t.} & X_{ii} = 1 \ (i = 1, \ldots, n) \\ & & X \succeq 0. \end{array} \tag{85}$$

Obviously, $m^*(A) \leq s^*(A)$. How well does the semidefinite bound $s^*(A)$ approximate $m^*(A)$? Obviously $m^*(A) = s^*(A)$ when all off-diagonal entries of A are nonnegative. We saw in Section 5.3 that $\frac{m^*(A)}{s^*(A)} \geq \alpha_0$ (the GW ratio from (77)) in the special case when A is the Laplacian matrix of a graph; that is, when $Ae = 0$ and $A_{ij} \leq 0$ for all $i \neq j$. (Note that these conditions imply that $A \succeq 0$.) Nesterov [149] studies the quality of the SDP relaxation for general A. When $A \succeq 0$ he shows the lower bound $\frac{2}{\pi}$ for the ratio $\frac{m^*(A)}{s^*(A)}$ and, based on this, he gives upper bounds for the relative accuracy

$s^*(A) - m^*(A)$ for indefinite A. The basic step consists in giving a trigonometric reformulation of the problem (84), analogue to the trigonometric reformulation (79) for max-cut.

Proposition 14. *Given a symmetric matrix A,*

$$\begin{array}{rll} m^*(A) = & \max & \frac{2}{\pi}\langle A, \arcsin(X)\rangle \\ & s.t. & X_{ii} = 1 \qquad (i = 1, \ldots, n) \\ & & X \succeq 0 \end{array} \tag{86}$$

setting $\arcsin(X) := (\arcsin(x_{ij}))_{i,j=1}^n$. *Moreover,* $m^*(A) \geq \frac{2}{\pi}s^*(A)$ *if* $A \succeq 0$.

Proof. Denote by μ the maximum of the program (86). Let x be an optimum solution to the program (84) and set $X := xx^T$. Then X is feasible for (86) with objective value $\frac{2}{\pi}\langle A, \arcsin(X)\rangle = \langle A, xx^T\rangle = m^*(A)$, which shows that $m^*(A) \leq \mu$. Conversely, let X be an optimum solution to (86) and let $v_1, \ldots, v_n$ be vectors such that $X_{ij} = v_i^T v_j$ for all i, j. Let r be a random unit vector. Then the expected value of $\text{sign}(r^T v_i)\text{sign}(r^T v_j)$ is equal to

$$1 - 2\ \text{prob}(\text{sign}(r^T v_i) \neq \text{sign}(r^T v_j)) = 1 - 2\frac{\arccos(v_i^T v_j)}{\pi} = \frac{2}{\pi}\arcsin(v_i^T v_j).$$

Therefore, the expected value E_A of $\sum_{i,j} a_{ij}\text{sign}(r^T v_i)\text{sign}(r^T v_j)$ is equal to $\frac{2}{\pi}\sum_{i,j} a_{ij}\arcsin(v_i^T v_j) = \frac{2}{\pi}\langle A, \arcsin(X)\rangle = \mu$. On the other hand, $\sum_{i,j} a_{ij}\text{sign}(r^T v_i)\text{sign}(r^T v_j) \leq m^*(A)$, since the vector $(\text{sign}(r^T v_i))_{i=1}^n$ is feasible for (84) for any unit vector r. This implies that $E_A \leq m^*(A)$ and thus $\mu \leq m^*(A)$. Assume $A \succeq 0$. Then, $\langle A, \arcsin(X)\rangle = \langle A, \arcsin(X) - X\rangle + \langle A, X\rangle \geq \langle A, X\rangle$, using the fact that $\arcsin(X) - X \succeq 0$ if $X \succeq 0$. Hence, $m^*(A) \geq \frac{2}{\pi}s^*(A)$ if $A \succeq 0$. ∎

Let $m_*(A)$ (resp. $s_*(A)$) denote the optimum value of the program (84) (resp. (85)) where we replace maximization by minimization. Applying the duality theorem for semidefinite programming, we obtain:

$$s^*(A) = \min(e^T y \mid \text{diag}(y) - A \succeq 0), \tag{87}$$

$$s_*(A) = \max(e^T z \mid A - \text{diag}(z) \succeq 0). \tag{88}$$

For $0 \leq \alpha \leq 1$, set $s_\alpha := \alpha s^*(A) + (1 - \alpha)s_*(A)$.

Lemma 15. *For* $\alpha := \frac{2}{\pi}$, $s_*(A) \leq m_*(A) \leq s_{1-\alpha} \leq s_\alpha \leq m^*(A) \leq s^*(A)$.

Proof. We show the inequality $m_*(A) \leq s_{1-\alpha}(A)$, that is, $s^*(A) - m_*(A) \geq \frac{2}{\pi}(s^*(A) - s_*(A))$. Let y (resp. z) be an optimum solution to (87) (resp. (88)). Then,

$$s^*(A) - m_*(A) = e^T y + m^*(-A) = m^*(\text{diag}(y) - A) \geq \frac{2}{\pi}s^*(\text{diag}(y) - A)$$

by Proposition 14, since $\text{diag}(y) - A \succeq 0$. To conclude, note that $s^*(\text{diag}(y) - A) = e^T y + s^*(-A) = e^T y - s_*(A) = s^*(A) - s_*(A)$. The inequality $s_\alpha(A) \leq m^*(A)$ can be shown similarily. ∎

The above lemma can be used for proving the following bounds on the relative accuracy $m^*(A) - s_\alpha$.

Theorem 16. *Set* $\alpha := \frac{2}{\pi}$ *and* $\beta := \frac{\alpha^2+2\alpha-1}{3\alpha-1}$. *Then,*

$$\frac{m^*(A) - s_\alpha}{m^*(A) - m_*(A)} \leq \frac{\pi}{2} - 1 < \frac{4}{7} \quad \textit{and} \quad \frac{|m^*(A) - s_\beta(A)|}{m^*(A) - m_*(A)} \leq \frac{\pi - 2}{6 - \pi} < \frac{2}{5}.$$

The above results can be extended to quadratic problems of the form:

$$\max \ x^T Ax \text{ subject to } [x]^2 \in F$$

where F is a closed convex set in $\mathbb{R}^n$ and $[x]^2 := (x_1^2, \ldots, x_n^2)$. See [191], chapter 13 in [194], [197], [200] for further results. Inapproximability results are given in [29].

6.2 Approximating the maximum bisection problem

The *maximum weight bisection problem* is a variant of the max-cut problem where one wants to find a cut $\delta(S)$ such that $|S| = \frac{n}{2}$ (a *bisection* or *equicut*) (n being assumed even) having maximum weight. This is an NP-hard problem, for which no approximation algorithm with a performance ratio $> \frac{16}{17}$ exists unless $P = NP$ [89]. Polynomial time approximation schemes are known to exist for this problem over dense graphs [15] and over planar graphs [95].

Extending the Goemans-Williamson approach to max-cut, Frieze and Jerrum [70] gave a randomized 0.651-approximation algorithm for the maximum weight bisection problem. Ye [198] improved the performance ratio to 0.6993 by combining the Frieze-Jerrum approach with some rotation argument applied to the optimum solution of the semidefinite relaxation. Halperin and Zwick [86] further improved the approximation ratio to 0.7016 by strengthening the SDP relaxation with the triangle inequalities. Details are given below.

Given a graph $G = (V, E)$ ($V = \{1, \ldots, n\}$) and edge weights $w \in \mathbb{R}_+^E$, the maximum weight bisection problem reads:

$$\begin{array}{ll} \max & \frac{1}{2}\sum_{ij \in E} w_{ij}(1 - x_i x_j) \\ \text{s.t.} & \sum_{i=1}^n x_i = 0 \\ & x_1, \ldots, x_n \in \{\pm 1\}. \end{array} \tag{89}$$

A natural semidefinite relaxation is:

$$\begin{array}{lll} W^* := & \max & \frac{1}{2}\sum_{ij \in E} w_{ij}(1 - X_{ij}) \\ & \text{s.t.} & X_{ii} = 1 \ (i \in V) \\ & & \langle J, X \rangle = 0 \\ & & X \succeq 0 \end{array} \tag{90}$$

The Frieze-Jerrum approximation algorithm:

1. *The SDP optimization phase:* Solve the SDP (90), let X be an optimum solution and let $v_1, \ldots, v_n$ be vectors such that $X_{ij} = v_i^T v_j$ for all i, j.

2. *The random hyperplane rounding phase:* Choose a random unit vector r and define the associated cut $\delta(S)$ where $S := \{i \in V \mid r^T v_i \geq 0\}$.

3. *Constructing a bisection:* Without loss of generality, assume that $|S| \geq \frac{n}{2}$. For $i \in S$, set $W(i) := \sum_{j \notin S} w_{ij}$. Order the elements of S as $i_1, \ldots, i_{|S|}$ in such a way that $W(i_1) \geq \ldots \geq W(i_{|S|})$ and define $\tilde{S} := \{i_1, \ldots, i_{\frac{n}{2}}\}$.

Then $\delta(\tilde{S})$ is a bisection whose weight satisfies

$$w(\delta(\tilde{S})) \geq \frac{n}{2|S|} w(\delta(S)). \tag{91}$$

Consider the random variables $W := w(\delta(S))$ and $C := |S|(n - |S|)$; W is the weight of the cut $\delta(S)$ in G while C is the number of pairs $(i,j) \in V^2$ that are cut by the partition $(S, V \setminus S)$ (that is, the cardinality of the cut $\delta(S)$ viewed as cut in the complete graph K_n). The analysis of the GW algorithm from Section 5.3 shows the following lower bounds for the expected values $E(W)$ and $E(C)$:

$$E(W) \geq \alpha_0 W^*, \tag{92}$$

$$E(C) \geq \alpha_0 C^* \tag{93}$$

where $C^* := \frac{n^2}{4}$. Define the random variable

$$Z := \frac{W}{W^*} + \frac{C}{C^*}. \tag{94}$$

Then, $Z \leq 2$ and $E(Z) \geq 2\alpha_0$.

Lemma 17. *If $Z \geq 2\alpha_0$ then $w(\delta(\tilde{S})) \geq 2\left(\sqrt{2\alpha_0} - 1\right) W^*$.*

Proof. Set $w(\delta(S)) = \lambda W^*$ and $|S| = \sigma n$. Then, $Z = \lambda + 4\sigma(1-\sigma) \geq 2\alpha_0$, implying $\lambda \geq 2\alpha_0 - 4\sigma(1-\sigma)$. Using (91), we obtain that

$$w(\delta(\tilde{S})) \geq \frac{n}{2|S|} w(\delta(S)) = \frac{\lambda W^*}{2\sigma} \geq W^* \frac{2\alpha_0 - 4\sigma(1-\sigma)}{2\sigma} \geq 2(\sqrt{2\alpha_0} - 1) W^*.$$

(The last inequality being a simple verification.) ∎

As $E(Z) \geq 2\alpha_0$, the strategy employed by Frieze and Jerrum in order to find a bisection satisfying the conclusion of Lemma 17 is to repeat the above steps 2 and 3 of the algorithm N times, where N depends on some small $\epsilon > 0$ ($N = \lceil \frac{1}{\epsilon} \ln \frac{1}{\epsilon} \rceil$) and to choose as output bisection the heaviest among the N bisections produced throughout the N runs. Then, with high probability, the largest among the variables Z produced throughout the N runs will be greater than or equal to $2\alpha_0$. Therefore, it follows from Lemma 17 that the weight of the output bisection is at least $(2(\sqrt{2\alpha_0} - 1) - \epsilon) W^*$. For ϵ small enough, this shows a performance ratio of 0.651.

Ye [198] shows an improved approximation ratio of 0.6993. For this, he modifies the Jerrum-Frieze algorithm in the following way. Instead of applying the random hyperplane rounding phase to the optimum solution X of (90), he applies it to the modified matrix $\rho X + (1-\rho) I$, where ρ is a parameter to be determined. This operation is analogue to the 'outward rotation' used by Zwick [202] for the max-cut problem and mentioned in Section 5.4.

The starting point is to replace relations (92) and (93) by

$$E(W) \geq \alpha W^* \text{ and } E(C) \geq \gamma C^* \tag{95}$$

where $\alpha = \alpha(\rho)$ and $\gamma = \gamma(\rho)$ are lower bounds to be determined on the ratios $\frac{E(W)}{W^*}$ and $\frac{E(C)}{C^*}$, respectively. In fact, the following choices can be made for α, γ:

$$\alpha(\rho) := \min_{-1 \leq x < 1} \frac{2}{\pi} \frac{\arccos(\rho x)}{1-x}, \tag{96}$$

$$\gamma(\rho) := \min_{-1 \leq x < 1} \frac{2}{\pi} \frac{\arccos(\rho x) - x \arccos \rho}{1-x}. \tag{97}$$

Indeed, $E(W) = \frac{1}{2} \sum_{ij \in E} w_{ij} \frac{2}{\pi} \arccos(\rho X_{ij}) \geq \alpha(\rho) W^*$. By the definition of $\gamma(\rho)$, $\frac{2}{\pi} \arccos(\rho x) \geq (1-x)\gamma(\rho) + \frac{2}{\pi} x \arccos \rho$ for $x \in [-1, 1]$. Therefore,

$$E(C) = \frac{1}{4} \sum_{i \neq j \in \{1, \ldots, n\}} \frac{2}{\pi} \arccos(\rho X_{ij}) \geq \frac{1}{4} \gamma(\rho) \sum_{i \neq j} (1 - X_{ij}) + \frac{1}{2\pi} \arccos \rho \sum_{i \neq j} X_{ij} = \frac{n^2}{4} \gamma(\rho) - \frac{\arccos \rho}{2\pi} n.$$

For n large enough, the linear term can be ignored and the result follows.

Modify the definition of Z from (94) as

$$Z := \frac{W}{W^*} + \tau \frac{C}{C^*}$$

where $\tau := \frac{\alpha}{2\beta} \left(\frac{1}{\sqrt{1-\beta}} - 1 \right)$. The proof of Lemma 17 can be adapted to show that, if $Z \geq \alpha + \tau\gamma$, then $E(w(\tilde{S})) \geq \frac{\alpha}{1+\sqrt{1-\beta}} W^*$. For $\rho = 0.89$, one can compute that $\alpha(\rho) \geq 0.8355$, $\gamma(\rho) \geq 0.9621$, and $\frac{\alpha}{1+\sqrt{1-\beta}} > 0.6993$. Therefore, this shows that Ye's algorithm is a 0.6993-approximation algorithm.

Halperin and Zwick [86] can improve the performance ratio to 0.7016. They achieve this by adding one more ingredient to Ye's algorithm; namely, they strengthen the SDP relaxation (90) by adding the triangle inequalities:

$$X_{ij} + X_{ik} + X_{jk} \geq -1, \; X_{ij} - X_{ik} - X_{jk} \geq -1 \; \text{ for distinct } i, j, k \in \{1, \ldots, n\}.$$

Although triangle inequalities had already been used earlier by some authors to obtain better approximations (e.g., in [64] for the max-cut problem in bounded degree graphs as mentioned in Section 5.4), they were always analyzed from a local point of view (e.g., in the above mentioned example, in a local search phase, searching for misplaced vertices). In contrast, Halperin and Zwick are able to make a global analysis of the contribution of triangle inequalities. Namely, they show that the function $\gamma(\rho)$ from (97) can be replaced by

$$\gamma'(\rho) := \min_{-1 \leq x \leq -\frac{1}{3}} \frac{1}{\pi} \left(\arccos(\rho x) + \frac{3(x+1)}{4} \arccos\left(-\frac{\rho}{3}\right) + \frac{1-3x}{4} \arccos \rho \right),$$

which enables them to demonstrate a better performance ratio (using appropriate values for the parameters ρ and τ). (Note that $\gamma'(\rho) > \gamma(\rho)$ for $0 < \rho < 1$.)

Let us give a flavor of how the function $\gamma'(\rho)$ comes up. The goal is to find a lower bound for the ratio $\frac{E(C)}{C^*} = \frac{4}{\pi n^2} \sum_{1 \le i < j \le n} \arccos(\rho X_{ij})$. Let A (resp. B, C) denote the set of pairs ij for which $X_{ij} < -\frac{1}{3}$ (resp. $-\frac{1}{3} \le X_{ij} \le 0$, $0 \le X_{ij} \le 1$). By the triangle inequalities, the graph on $\{1, \dots, n\}$ with edge set A is triangle free, which implies that $|A| \le \frac{n^2}{4}$. Thus the optimum value of the following nonlinear program is a lower bound for $\frac{E(C)}{C^*}$:

$$\begin{array}{ll} \min & \frac{4}{\pi n^2} \sum_{i<j} \arccos(\rho z_{ij}) \\ \text{s.t.} & \quad \sum_{i<j} z_{ij} = -\frac{n}{2} \\ & -1 \le z_{ij} \le 1 \ (i<j) \\ & |\{ij \mid z_{ij} < -\frac{1}{3}\}| \le \frac{n^2}{4}. \end{array}$$

Halperin and Zwick show then that the above minimum can be expressed in closed form as $\gamma'(\rho)$.

Feige, Karpinski and Langberg [65] design a 0.795-approximation algorithm for the maximum bisection problem restricted to regular graphs. One of their key results is the following: Given a cut $\delta(S)$ in a regular graph G, one can efficiently construct a bisection $\delta(S')$ whose weight is at least $0.9027\ w(\delta(S))$. Hence, if we start with the cut $\delta(S)$ given as output of the Goemans-Williamson algorithm, then this gives an approximation algorithm with performance ratio $0.9027 \cdot 0.878 \sim 0.793$; a further improvement is demonstrated in [65].

Extensions to variations of the bisection problem. The following variations of the bisection problem have been studied in the literature: (i) the *maximum $\frac{n}{2}$-vertex cover problem,* (ii) the *maximum $\frac{n}{2}$-dense subgraph problem,* (iii) the *maximum $\frac{n}{2}$-uncut problem*, which ask for a subset $S \subseteq V$ of size $\frac{n}{2}$ maximizing the total weight of the edges incident to S, contained in S, contained in S or its complement, respectively. Halperin and Zwick [86] treat these three problems (together with the maximum bisection problem as well as some directed analogues) in a unified framework and they can show the best approximation ratios known up to today, namely, 0.8452 for problem (i), 0.6221 for problem (ii) and 0.6436 for problem (iii).

6.3 Approximating the max k-cut problem

Given a graph $G = (V, E)$, edge weights $w \in \mathbb{R}_+^E$ and an integer $k \ge 2$, the *max k-cut problem* asks for a partition $\mathcal{P} = (S_1, \dots, S_k)$ of V whose weight $w(\mathcal{P}) := \sum_{1 \le h < h' \le k} \sum_{ij \in E | i \in S_h, j \in S_{h'}} w_{ij}$ is maximum. The set of edges whose end nodes belong to distinct classes of the partition is a *k-cut*, denoted as $\delta(S_1, \dots, S_k)$. For $k = 2$, we find the max-cut problem. For any $k \ge 2$, the max k-cut problem is NP-hard; moreover, there can be no polynomial time approximation algorithm for it with performance ratio $1 - \frac{1}{239k}$, unless P=NP [99].

A simple heuristic for max k-cut is to partition V randomly into k sets. As the probability that two nodes fall in the same class is $\frac{1}{k}$, the expected weight of the k-cut produced in this way is $\sum_{ij \in E} w_{ij}(1 - \frac{1}{k}) \ge w(E)(1 - \frac{1}{k})$ and, therefore, the simple random partition heuristic has a performance guarantee of $1 - \frac{1}{k}$.

Frieze and Jerrum [70] present an approximation algorithm for max k-cut with performance guarantee α_k satisfying

(i) $\alpha_k > 1 - \frac{1}{k}$ and $\lim_{k \to \infty} \frac{\alpha_k - (1 - \frac{1}{k})}{2k^{-2} \ln k} = 1$,

(ii) $\alpha_2 = \alpha_0 \ge 0.878567$ (recall (77)), $\alpha_3 \ge 0.832718$, $\alpha_4 \ge 0.850304$, $\alpha_5 \ge 0.874243$, $\alpha_{10} \ge 0.926642$, $\alpha_{100} \ge 0.990625$.

In particular, the Frieze-Jerrum algorithm has a better performance guarantee than the simple random heuristic.

One can model the max k-cut problem on a graph $G = (V, E)$ ($V = \{1, \ldots, n\}$) by having n variables $x_1, \ldots, x_n$ taking one of k possible values. For $k = 2$ the 2 possible values are ± 1 and for $k \geq 2$ one can choose as possible values a set of k unit vectors $a_1, \ldots, a_k \in \mathbb{R}^{k-1}$ satisfying

$$a_i^T a_j = -\frac{1}{k-1} \text{ for } 1 \leq i \neq j \leq k.$$

(Such vectors exist since the matrix $\frac{k}{k-1} I_k - \frac{1}{k-1} J_k$ is positive semidefinite.) Hence the max k-cut problem can be formulated as

$$\begin{array}{rll} \mathrm{mc}_k(G, w) := & \max & \frac{k-1}{k} \sum_{ij \in E} w_{ij}(1 - x_i^T x_j) \\ & \text{s.t.} & x_1, \ldots, x_n \in \{a_1, \ldots, a_k\} \end{array} \tag{98}$$

and the following is a semidefinite relaxation of (98):

$$\begin{array}{rlll} \mathrm{sdp}_k(G, w) := & \max & \frac{k-1}{k} \sum_{ij \in E} w_{ij}(1 - X_{ij}) & \\ & \text{s.t.} & X_{ii} = 1 & (i \in V) \\ & & X_{ij} \geq -\frac{1}{k-1} & (i \neq j \in V) \\ & & X \succeq 0. & \end{array} \tag{99}$$

The Frieze-Jerrum approximation algorithm for max k-cut:

1. Solve (99) to obtain unit vectors $v_1, \ldots, v_n$ satisfying $v_i^T v_j \geq -\frac{1}{k-1}$ $(i, j \in V)$ and $\mathrm{sdp}_k(G, w) = \frac{k-1}{k} \sum_{ij \in E} w_{ij}(1 - v_i^T v_j)$.
2. Choose k independent random vectors $r_1, \ldots, r_k \in \mathbb{R}^n$. (This can be done by chosing their kn components as independant random variables from the standard normal distribution with mean 0 and variance 1.)
3. Partition V into $S_1, \ldots, S_k$ where S_h consists of the nodes $i \in V$ for which $v_i^T r_h = \max_{h'=1,\ldots,k} v_i^T r_{h'}$. (Break ties arbitrarily as they occur with probability 0.)

When $k = 2$ the algorithm reduces to the Goemans-Williamson algorithm for max-cut. Given two unit vectors $u, v \in \mathbb{R}^n$, the probability that $\max_{1 \leq h \leq k} u^T r_h$ and $\max_{1 \leq h \leq k} v^T r_h$ are both attained by the same vector within $r_1, \ldots, r_k$ depends only on the angle between u and v, i.e., on $\rho := u^T v$, and it is equal to $k \cdot \mathrm{prob}(u^T r_1 = \max_{1 \leq h \leq k} u^T r_h \text{ and } v^T r_1 = \max_{1 \leq h \leq k} v^T r_h)$; denote this probability as $kI(\rho)$. Then the expected weight of the k-cut $\delta(S_1, \ldots, S_k)$ produced by the Frieze-Jerrum algorithm is equal to

$$\begin{array}{l} \sum_{ij \in E} w_{ij} \mathrm{prob}(ij \in \delta(S_1, \ldots, S_k)) = \sum_{ij \in E} w_{ij}(1 - kI(v_i^T v_j)) \\ = \sum_{ij \in E} w_{ij} \left(\frac{k}{k-1} \frac{1 - kI(v_i^T v_j)}{1 - v_i^T v_j} \right) \left(\frac{k-1}{k}(1 - v_i^T v_j) \right) \geq \alpha_k \ \mathrm{sdp}_k(G, w), \end{array}$$

setting

$$\alpha_k := \min_{-\frac{1}{k-1} \leq \rho < 1} \frac{k}{k-1} \frac{1 - kI(\rho)}{1 - \rho}. \tag{100}$$

For $k = 2$, $\alpha_2 = \alpha_0$ can be computed exactly. For $k \geq 3$, the evaluation of α_k is more complicated and relies on the computation of the function $I(\rho)$ which can be expressed as a multiple integral. Using a

Taylor series expansion for $I(\rho)$, Frieze and Jerrum could show the lower bounds for α_k mentioned at the beginning of this subsection.

For $k = 3$, de Klerk and Pasechnik [109] give a closed form expression for $I(\rho)$ which enables them to show that

$$\alpha_3 = \frac{7}{12} + \frac{3}{4\pi^2} \arccos^2(-1/4).$$

Thus $\alpha_3 > 0.836008$ (instead of the lower bound 0.832718 of Frieze and Jerrum). Goemans and Williamson [78] find the same expression for α_3 using another formulation for max 3-cut based on complex semidefinite programming.

De Klerk, Pasechnik and Warners [108] prove a better lower bound for α_k for small $k \geq 3$. For instance, they show that $\alpha_4 \geq 0.857487$ (instead of 0.850304). For this they present another approximation algorithm for max k-cut (equivalent to the Frieze-Jerrum algorithm for the graphs G with $\vartheta(\overline{G}) \leq k$) which enables them to reformulate the function $I(\rho)$ in terms of the volume of a spherical simplex and do more precise computations.

6.4 Approximating graph colouring

Determining the chromatic number of a graph is a hard problem. Lund and Yannakakis [137] show that there is a constant $\epsilon > 0$ for which there exists no polynomial algorithm which can colour any graph G using at most $n^\epsilon \chi(G)$ unless P=NP. Khanna, Linial and Safra [106] show that it is not possible to colour a 3-colourable graph with 4 colours in polynomial time unless P=NP.

On the positive side, Wigderson [193] shows that it is possible to colour in polynomial time a 3-colourable graph with $3\lceil\sqrt{n}\rceil$ colours and, more generally, a k-colourable graph with $2kn^{1-\frac{1}{k-1}}$ colours; we will come back to this result later in this section. Later Blum [32] gives a polynomial time algorithm colouring a 3-colourable graph with $O(n^{\frac{3}{8}} \log^{\frac{8}{5}} n)$. Using semidefinite programming and randomized rounding, Karger, Motwani and Sudan [100] could design a better approximation algorithm. Namely, they present a randomized polynomial time algorithm which colours a 3-colourable graph with maximum degree Δ with $O(\Delta^{\frac{1}{3}}\sqrt{\log \Delta}\log n)$ or $O(n^{\frac{1}{4}}\sqrt{\log n})$ colours and, more generally, a k-colourable graph with $O(\Delta^{1-\frac{2}{k}}\sqrt{\log \Delta}\log n)$ or $O(n^{1-\frac{3}{k+1}}\sqrt{\log n})$ colours.

We will show below a weaker version of the Karger-Motwani-Sudan result; namely, how to find a $O(n^{0.387})$ colouring for a 3-colourable graph.

The first step in the Karger-Motwani-Sudan algorithm consists in solving a semidefinite relaxation for the colouring problem. We saw in Section 4.2 that the theta numbers $\vartheta(\overline{G})$ and $\vartheta'(\overline{G})$ constitute lower bounds for the chromatic number of G. Karger, Motwani and Sudan consider the SDP program (62) defining $\vartheta'(\overline{G})$ as SDP relaxation for the colouring problem and they introduce the notion of vector colouring. A *vector k-colouring* of G is an assignment of vectors $v_1, \ldots, v_n$ to the nodes of G such that $v_i^T v_j \leq -\frac{1}{k-1}$ for every edge $ij \in E$. Then the *vector colouring number* $\chi_v(G)$ is defined as the smallest $k \geq 2$ for which there exists a vector k-colouring. By the discussion above, $\chi_v(G) = \vartheta'(\overline{G})$; if in the definition of vector colouring one requires that the inequalities $v_i^T v_j \leq -\frac{1}{k-1}$ hold at equality for all edges, then we obtain the *strict vector colouring number* which coincides with $\vartheta(\overline{G})$. Let us point out that the gap between the vector chromatic number and the chromatic number can be arbitrarily large. Indeed, a class of graphs is constructed in [100] having $\chi_v(G) = 3$ while $\chi(G) \geq n^{0.0113}$. Moreover, Feige [61] shows that for all $\epsilon > 0$ there exist families of graphs with $\chi(G) \geq \vartheta(\overline{G})n^{1-\epsilon}$.

The hard part in the Karger-Motwani-Sudan algorithm consists of constructing a good proper colouring from a vector k-colouring. There are two steps: first construct a semicolouring and then from it a proper colouring. A *k-semicolouring* of a graph on n nodes being an assignment of k colours to at least half of the nodes in such a way that no two adjacent nodes receive the same colour. This is a useful notion, as an algorithm for semicolouring yields an algorithm for proper colouring.

Lemma 18. *Let $f : \mathbb{Z}_+ \longrightarrow \mathbb{Z}_+$ be a monotone increasing function. If there is a randomized polynomial time algorithm which $f(i)$-semicolours every i-vertex subgraph of graph G, then this algorithm can colour G with $O(f(n)\log n)$ colours. Moreover, if there exists some $\epsilon > 0$ such that $f(i) = O(i^\epsilon)$ for all i, then the algorithm can colour G with $f(n)$ colours.*

PROOF. We show how to colour any p-vertex subgraph H of G. By assumption one can semicolour H with $f(p)$ colours. Let S denote the set of nodes of H that have not been coloured; then $|S| \leq \frac{p}{2}$. One can recursively colour the subgraph of H induced by S using a new set of colours.

Let $c(p)$ denote the maximum number of colours that the above algorithm needs for colouring an arbitrary p-vertex subgraph of G. Then,

$$c(p) \leq c\left(\frac{p}{2}\right) + f(p).$$

This recurrence relation implies that $c(p) = O(f(p)\log p)$. Moreover, if $f(p) = p^\epsilon$, one can easily verify that $c(p) = O(f(p))$. ▮

In view of Lemma 18, we are now left with the task of transforming a vector k-colouring into a good semicolouring.

Theorem 19. *Every vector 3-colourable graph G with maximum degree Δ has a $O(\Delta^{\log_3 2})$-semicolouring which can be constructed in polynomial time with high probability.*

PROOF. Let $v_1, \ldots, v_n \in \mathbb{R}^n$ be unit vectors forming a vector 3-colouring of G, i.e., $v_i^T v_j \leq -\frac{1}{2}$ for all edges $ij \in E$; this means that the angle between v_i and v_j is at least $\frac{2\pi}{3}$ for all edges $ij \in E$. Choose independently N random hyperplanes. This induces a partition of the space $\mathbb{R}^n$ into 2^N regions and one colours the nodes of G with 2^N colours depending in which region their associated vectors v_i are located. Then the probability that an edge is monochromatic is at most 3^{-N} and thus the expected number of monochromatic edges is at most $|E|3^{-N} \leq \frac{1}{2}n\Delta 3^{-N}$. By Markov's inequality, the probability that the number of monochromatic edges is more than twice the expected number is at most $\frac{1}{2}$. After repeating the process t times, we find with probability $\geq 1 - \frac{1}{2^t}$ a colouring of G for which the number of monochromatic edges is at most $n\Delta 3^{-N}$. Setting $N := 2 + \lceil \log_3 \Delta \rceil$, we have $n\Delta 3^{-N} \leq \frac{n}{4}$. As the number of nodes that are incident to a monochromatic edge is $\leq \frac{n}{2}$, we have found a semicolouring using $2^N \leq 8\Delta^{\log_3 2}$ colours. ▮

As $\log_3 2 < 0.631$, Theorem 19 and Lemma 18 imply a colouring with $O(n^{0.631})$ colours. This is yet weaker than Wigderson's $O(n^{0,5})$-colouring algorithm. In fact, the result can be improved using the following idea of Wigderson.

Theorem 20. *There is a polynomial time algorithm which, given a 3-colourable graph G and a constant $\delta \leq n$, finds an induced subgraph H of G with maximum degree $\Delta_H < \delta$ and a $\frac{2n}{\delta}$ colouring of*

$G \backslash H$.

Proof. If G has a node v of degree $\geq \delta$, colour the subgraph induced by $N(v)$ with two colours and delete $\{v\} \cup N(v)$ from G. We repeat this process using two new colours at each deleted neighborhood and stop when we arrive at a graph H whose maximum degree is less than δ. ▮

Applying Theorem 20 with $\delta = \sqrt{n}$ and the fact that a graph with maximum degree Δ has a $(\Delta + 1)$-colouring, one finds Wigderson's polynomial algorithm for colouring a 3-colourable graph with $3\lceil\sqrt{n}\rceil$ colours.

Theorem 21. *A 3-colourable graph can be coloured with $O(n^{0.387})$ colours by a polynomial time randomized algorithm.*

Proof. Let G be a 3-colourable graph. Applying Theorem 20 with $\delta := n^{0.613}$, we find an induced subgraph H of maximum degree $\Delta_H < \delta$ and a colouring of $G \backslash H$ using $\frac{2n}{\delta} = O(n^{0.387})$ colours. By Theorem 19 and Lemma 18, H can be coloured with $O(\delta^{\log_3 2}) = O(n^{0.387})$ colours. This shows the result. ▮

In order to achieve the better $O(n^{\frac{1}{4}}\sqrt{\log n})$ colouring for a 3-colourable graph, one has to improve Theorem 19 and show a $O(\Delta^{\frac{1}{3}}\sqrt{\log \Delta})$-semicolouring. For this, Karger, Motwani and Sudan use another randomized technique for constructing a colouring from a vector colouring whose analysis is more sophisticated, that we do not describe here.

6.5 Approximating the maximum stable set and vertex cover problems

Determining the stability number of a graph is a hard problem. Arora et al. [16] show the existence of a constant $\epsilon > 0$ for which there is no polynomial time algorithm permitting to find a stable set in a graph G of size at least $n^{-\epsilon}\alpha(G)$ unless P=NP. We saw in Section 4.2 that the theta number $\vartheta(G)$ is a polynomially computable upper bound for $\alpha(G)$ which is tight for perfect graphs, in which case a maximum cardinality stable set can be found in polynomial time. For general graphs, the gap between $\alpha(G)$ and $\vartheta(G)$ can be arbitrarily large. Indeed, Feige [61] shows that for all $\epsilon > 0$ there is a family of graphs for which $\vartheta(G) > n^{1-\epsilon}\alpha(G)$. The proof of Feige is nonconstructive; Alon and Kahale [6] give the following constructive proof for this result.

Theorem 22. *For every $\epsilon > 0$ one can construct a family of graphs on n nodes for which $\vartheta(G) \geq (\frac{1}{2} - \epsilon)n$ and $\alpha(G) = O(n^{\delta})$ where $0 < \delta < 1$ is a constant depending on ϵ.*

Proof. Given integers $0 < s < q$, let G_{qs} denote the graph on $n = \binom{2q}{q}$ nodes corresponding to all subsets A of $Q := \{1, \ldots, 2q\}$ with cardinality $|A| = q$, where A, B are adjacent if $|A \cap B| = s$. We begin with evaluating the theta number of G_{qs}. For every vertex A of G_{qs}, set $d_A := (x+1)\chi^A - \chi^Q$, where x is the largest root of the quadratic polynomial $sx^2 - 2(q-s)x + s = 0$. Then, $d_A^T d_B = 0$ for all adjacent A, B. Therefore, the vectors $v_A := \frac{d_a}{\|d_A\|}$ form an orthonormal representation of $\overline{G}_{qs}$. Setting

$d := \frac{1}{\sqrt{2q}}(1,\ldots,1)^T$ and using the definition from Theorem 11, we obtain:

$$\vartheta(G_{qs}) \geq \sum_A (d^T v_A)^2 = n\frac{(x-1)^2}{2(x^2+1)} = \frac{n}{2}\frac{q-2s}{q-s}.$$

In order to evaluate the stability number of G_{qs}, one can use the following result of Frankl and Rödl [69]: For every $\gamma > 0$, there exists $0 < \delta < 1$ for which $\alpha(G_{qs}) \leq n^\delta$ if $\gamma q < s < (1-\gamma)q$.

We now indicate how to choose the parameters q, s in order to achieve the conclusion of the theorem. Let $\epsilon > 0$ be given. Define s as the largest integer for which $s < \frac{q}{2}$ and $\frac{q-2s}{2(q-s)} > \frac{1}{2} - \epsilon$ (i.e., $s < \frac{2q\epsilon}{1+2\epsilon}$). Choose γ such that $0 < \gamma < \frac{s}{q}$. Then $\gamma q < s < (1-\gamma)q$ and thus $\alpha(G_{qs}) \leq n^\delta$ for some $0 < \delta < 1$ by the Frankl-Rödl result. ▮

On the positive side, Alon and Kahale [6] show the following two results; we present the second one without proof.

Theorem 23. *Let G be a graph on n nodes, $k \geq 3$, $m \geq 1$ be integers. If $\vartheta(G) \geq \frac{n}{k} + m$, then an independent set of cardinality $\Omega(m^{\frac{3}{k+1}} \log^{-\frac{1}{2}} m)$ can be found in randomized polynomial time.*

Proof. Using the definition of $\vartheta(G)$ from Theorem 11, there exist unit vectors $d, v_1, \ldots, v_n$ where $v_1, \ldots, v_n$ form an orthonormal representation of $\overline{G}$. These vectors can be found in polynomial time since, as the proof of Theorem 11 shows, they can be computed from an optimum solution to the SDP program (53). Order the nodes in such a way that $(d^T v_1)^2 \geq \ldots \geq (d^T v_n)^2$. As $\vartheta(G) \geq \frac{n}{k} + m$ and $(d^T v_i)^2 \leq 1$ for all i, we have $(d^T v_m)^2 \geq \frac{1}{k}$. Let H denote the subgraph of G induced by the nodes $1, \ldots, m$. Then, $v_1, \ldots, v_m$ is an orthonormal representation of $\overline{H}$, the complementary graph of H. Using the definition of the theta number from Theorem 13, we deduce that

$$\vartheta(\overline{H}) \leq \max_{i=1,\ldots,m} \frac{1}{(d^T v_i)^2} \leq k.$$

Therefore, H has a vector k-colouring. Applying the Karger-Motwani-Sudan results from the preceding subsection, one can find in randomized polynomial time a $O(m^{1-\frac{3}{k+1}}\sqrt{\log m})$ colouring of H. Then the largest colour class in this colouring has cardinality $\Omega(m^{\frac{3}{k+1}} \log^{-\frac{1}{2}} m)$. ▮

Theorem 24. *If G is a graph on n nodes such that $\vartheta(G) > Mn^{1-\frac{2}{k}}$ for an appropriate absolute constant M, one can find in polynomial time a stable set in G of cardinality k.* ▮

We now turn to the vertex cover problem. A subset $X \subseteq V$ is a *vertex cover* if every edge is adjacent to a node in X; that is, if $V \setminus X$ is a stable set. Denote by $vc(G)$ the minimum cardinality of a vertex cover in G. Thus $vc(G) = n - \alpha(G)$ and determining $vc(G)$ is therefore an NP-hard problem.

It is well known that $vc(G)$ can be approximated to within a factor of 2 in polynomial time using an LP relaxation of the problem. Indeed, consider the LP problem:

$$\begin{array}{lll} lp(G) := & \min & \sum_{i\in V} x_i \\ & \text{s.t.} & x_i + x_j \geq 1 \quad (ij \in E) \\ & & 0 \leq x_i \leq 1 \quad (i \in V) \end{array} \tag{101}$$

which is a linear relaxation of the vertex cover problem:

$$\begin{array}{lll} vc(G) := & \min & \sum_{i\in V} x_i \\ & \text{s.t.} & x_i + x_j \geq 1 \quad (ij \in E) \\ & & x_i \in \{0,1\} \quad (i \in V). \end{array} \tag{102}$$

Obviously, $lp(G) \leq vc(G)$. Moreover, $vc(G) \leq 2 \cdot lp(G)$; indeed, given an optimum solution x to (101), the set $X := \{i \in V \mid x_i \geq \frac{1}{2}\}$ is a vertex cover whose cardinality satisfies $|I| \leq 2 \cdot lp(G)$.

The existence of a polynomial time approximation algorithm for the vertex cover problem with performance ratio $2 - \epsilon$ is open for any $\epsilon > 0$. A natural idea for finding a better approximation algorithm is to try using a semidefinite relaxation, the most natural candidate being the theta number; more precisely, the quantity $n - \vartheta(G)$ which constitutes a lower bound for $vc(G)$. A first observation is that this SDP bound is at least as good as the LP bound; namely,

$$n - \vartheta(G) \geq lp(G).$$

To see it, use the definition from Theorem 11. let d be a unit vector and $v_1, \ldots, v_n$ an orthonormal representation of $\overline{G}$ such that $\vartheta(G) = \sum_{i\in V}(d^T v_i)^2$. Set $x_i := 1 - (d^T v_i)^2$ $(i \in V)$. Then x is a feasible solution to the program (101) which shows that $lp(G) \leq \sum_i x_i = n - \vartheta(G)$.

Kleinberg and Goemans [107] construct a class of graphs G for which the ratio $\frac{vc(G)}{n-\vartheta(G)}$ converges to 2 as n goes to infinity, which shows that no improvement is made by using SDP instead of LP. In fact, the class of graphs constructed in Theorem 22 displays the same behaviour.

6.6 Approximating MAX SAT

An instance of the MAX SAT problem in the Boolean variables $x_1, \ldots, x_n$ is composed of a collection $\mathcal{C}$ of clauses C with nonnegative weights w_C associated to them. Each *clause* C is of the form $z_1 \vee \ldots \vee z_k$ where each z_j is either a variable x_i or its negation $\overline{x}_i$ (called a *literal*); k is its *length* and C is satisfied if at least one of the literals $z_1, \ldots, z_k$ is assigned value 1 (if a variable x_i is assigned value 1 then its negation $\overline{x}_i$ is assigned value 0 and vice versa). The MAX SAT problem consists of finding an assignment of 0/1 values to the variables $x_1, \ldots, x_n$ so that the total weight of the satisfied clauses is maximized. Given an integer $k \geq 1$, the MAX kSAT problem is the special instance of MAX SAT where each clause has length at most k and MAX EkSAT is the instance where all clauses have length exactly k; an instance of MAX SAT is said to be *satisfiable* if there is an assignment of the x_i's satisfying all its clauses.

The MAX SAT and MAX kSAT problems are NP-hard. Moreover, Håstad [89] proved that, for any $\epsilon > 0$, there is no $(\frac{7}{8} + \epsilon)$-approximation algorithm for MAX SAT, unless P=NP; his result also holds when restricted to satisfiable instances of MAX E3SAT.

A $\frac{3}{4}$-approximation algorithm for MAX SAT. The first approximation algorithm for MAX SAT is the following $\frac{1}{2}$-approximation algorithm due to Johnson [98]. Given $p_i \in [0,1]$ $(i = 1, \ldots, n)$, set independently and randomly each variable x_i to 1 with probability p_i. Then the probability that a clause $C := \vee_{i\in I_C^+} x_i \bigvee \vee_{i\in I_C^-} \overline{x}_i$ is satisfied is equal to $1-\prod_{i\in I_C^+}(1-p_i)\prod_{i\in I_C^-} p_i$. If we set all p_i's to $\frac{1}{2}$, then the total expected weight $\hat{W}_1$ of satisfied clauses satisfies:

$$\hat{W}_1 = \sum_{C\in\mathcal{C}} w_C(1-\frac{1}{2^{k_C}}) \geq \frac{1}{2}\sum_{C\in\mathcal{C}} w_C$$

where k_C is the length of clause C. Therefore, this gives a randomized $\frac{1}{2}$-approximation algorithm for MAX SAT or a $(1-2^{-k})$-approximation algorithm for instances of MAX SAT where all clauses have length $\geq k$ (thus with performance ratio $\frac{3}{4}$ for MAX E2SAT and $\frac{7}{8}$ for MAX E3SAT); it can be derandomized using the method of conditional probabilities.

Goemans and Wiliamson [76] give an improved $\frac{3}{4}$-approximation algorithm using linear programming. Consider the integer programming problem:

$$\begin{array}{ll} \max & \sum_{C\in\mathcal{C}} w_C z_C \\ \text{s.t.} & z_C \leq \sum_{i\in I_C^+} y_i + \sum_{i\in I_C^-}(1-y_i)\ (C\in\mathcal{C}) \\ & 0\leq z_C\leq 1\ (C\in\mathcal{C}) \\ & y_i\in\{0,1\}\ (i=1,\ldots,n) \end{array} \tag{103}$$

and let Z^*_{LP} denote the optimum value of its linear programming relaxation obtained by relaxing the condition $y_i \in \{0,1\}$ by $0\leq y_i\leq 1$. If (y,z) is an optimum solution to (103), letting $x_i = 1$ if and only if $y_i = 1$, then clause C is satisfied precisely when $z_C = 1$; hence (103) solves the MAX SAT problem. The GW approximation algorithm goes as follows. First solve the LP relaxation of (103) and let (y,z) be an optimum solution to it. Then apply the Johnson's algorithm using the probabilities $p_i := y_i$; that is, set x_i to 1 with probability y_i. Setting $\beta_k := 1-\left(1-\frac{1}{k}\right)^k$ and using the fact that $1-\prod_{i\in I_C^+}(1-y_i)\prod_{i\in I_C^-} y_i \geq \beta_{k_C} z_C$, we find that the expected weight $\hat{W}_2$ of satisfied clauses satisfies:

$$\hat{W}_2 = \sum_{C\in\mathcal{C}} w_C\left(1-\prod_{i\in I_C^+}(1-y_i)\prod_{i\in I_C^-} y_i\right) \geq \sum_{C\in\mathcal{C}} w_C z_C \beta_{k_C}.$$

As β_k is decreasing with k, this gives a randomized β_k-approximation algorithm for instances of MAX SAT where all clauses have at most k literals; thus a $(1-\frac{1}{e})$-approximation algorithm for MAX SAT, since $\lim_{k\longrightarrow\infty}(1-\frac{1}{k})^k = \frac{1}{e}$.

In order to obtain the promised $\frac{3}{4}$ performance ratio, it suffices to combine the above two algorithms. For this, note that $\frac{1}{2}(1-\frac{1}{2^k}+\beta_k)\geq\frac{3}{4}$ for all $k\geq 1$. Therefore, $\frac{1}{2}(\hat{W}_1+\hat{W}_2)\geq\frac{3}{4}Z^*_{LP}$. Hence the following is a $\frac{3}{4}$-approximation algorithm for MAX SAT: with probability $\frac{1}{2}$, use the probabilities $p_i := \frac{1}{2}$ for determining the variables x_i and, with probability $\frac{1}{2}$, use instead the probabilities $p_i := y_i$.

Better approximation algorithms can be obtained using semidefinite relaxations instead of linear ones combined with adequate rounding techniques, as we now see.

The Goemans-Williamson α_0-approximation algorithm for MAX 2SAT and their 0.7554-approximation algorithm for MAX SAT. Using a semidefinite relaxation for MAX SAT instead of a linear one and the hyperplane rounding technique, one can show a better approximation algorithm. It is convenient to introduce the new Boolean variables $x_{n+i} = \overline{x}_i$ for $i=1,\ldots,n$. Then a clause C can be expressed as a disjunction $C = \bigvee_{i\in I_C} x_i$, of the variables $x_1,\ldots,x_{2n}$, with $I_C\subseteq\{1,\ldots,2n\}$. It is also convenient to work with ±1 variables v_i (instead of $y_i\in\{0,1\}$) and to introduce an additional ±1

variable v_0, the convention being to set x_i to 1 if $v_i = -v_0$ and to 0 if $v_i = v_0$. Hence the formulation (103) of MAX SAT can be rewritten as

$$\begin{array}{ll} \max & \sum_{C\in\mathcal{C}} w_C z_C \\ \text{s.t.} & z_C \leq \sum_{i\in I_C} \frac{1-v_0\cdot v_i}{2} \quad (C\in\mathcal{C}) \\ & 0 \leq z_C \leq 1 \quad (C\in\mathcal{C}) \\ & v_i \cdot v_{n+i} = -1 \quad (i=1,\ldots,n) \\ & v_0, v_1, \ldots, v_{2n} \in \{\pm 1\}. \end{array} \qquad (104)$$

For each clause $C = x_i \vee x_j$ of length 2, one can add the constraint:

$$z_C \leq 1 - \left(\frac{1+v_0\cdot v_i}{2}\right)\left(\frac{1+v_0\cdot v_j}{2}\right) = \frac{3 - v_0\cdot v_i - v_0\cdot v_j - v_i\cdot v_j}{4} \qquad (105)$$

which, in fact, implies the constraint $z_C \leq \frac{1-v_0\cdot v_i}{2} + \frac{1-v_0\cdot v_j}{2}$.

Let (SDP) denote the semidefinite relaxation of the program (104) augmented with the constraints (105) for all clauses of length 2, which is obtained by introducing a matrix variable $X = (X_{ij})_{i,j=0}^{2n} \succeq 0$ and replacing each product $v_i \cdot v_j$ by X_{ij}. In other words, this amounts to replacing the constraint $v_0, \ldots, v_{2n} \in \{\pm 1\}$ by the constraint $v_0, \ldots, v_{2n} \in S^n$, S^n being the unit sphere in $\mathbb{R}^{n+1}$ (the product $v_i \cdot v_j$ meaning then the inner product $v_i^T v_j$).

Goemans and Williamson [77] show that their basic α_0-approximation algorithm for max-cut extends to MAX 2SAT. Namely, solve the relaxation (SDP) and let $v_0, \ldots, v_n$ be the optimum unit vectors solving it; select a random unit vector r and let H_r be the hyperplane with normal vector r; set x_i to 1 if the hyperplane H_r separates v_0 and v_i and to 0 otherwise. Let θ_{ij} denote the angle (v_i, v_j). Then the probability prob(v_0, v_i) that the clause x_i is satisfied is equal to the probability that H_r separates v_0 and v_i and thus

$$\text{prob}(v_0, v_i) = \frac{\theta_{0i}}{\pi};$$

the probability prob(v_0, v_i, v_j) that the clause $x_i \vee x_j$ is satisfied is equal to the probability that a random hyperplane separates v_0 from at least one of v_i and v_j which can be verified to be equal to

$$\text{prob}(v_0, v_i, v_j) = \frac{1}{2\pi}(\theta_{0i} + \theta_{0j} + \theta_{ij})$$

using the inclusion/exclusion principle. Therefore, for a clause $C = x_i \vee x_j$, we have

$$\frac{\text{prob}(v_0, v_i, v_j)}{z_C} \geq \frac{2}{\pi}\frac{\theta_{0i} + \theta_{0j} + \theta_{ij}}{3 - \cos\theta_{0i} - \cos\theta_{0j} - \cos\theta_{ij}} \geq \alpha_0,$$

where $\alpha_0 \simeq 0.87856$ is the Goemans-Williamson ratio from (77). The above relation also holds when $i = j$, i.e., when C is a clause of length 1, in which case one lets prob(v_0, v_i, v_j) = prob(v_0, v_i). Hence the expected total weight of satisfied clauses is greater than or equal to α_0 times the optimum value of the relaxation (SDP); this gives therefore an α_0-approximation algorithm for MAX 2SAT.

This improved MAX 2SAT algorithm leads to a slightly improved 0.7554-approximation algorithm for general MAX SAT. For this, one considers the following three algorithms: (1) set x_i to 1 independently with probability $p_i := \frac{1}{2}$; (2) set x_i to 1 independently with probability $p_i := \frac{1-v_0^T v_i}{2}$; (3) select a random hyperplane H_r and set x_i to 1 if H_r separates v_0 and v_i (the v_i's being the optimum vectors to the relaxation (SDP)). One chooses algorithm (i) with probability q_i where $q_1 = q_2 = 0.4785$ and $q_3 = 1 - q_1 - q_2 = 0.0430$. Then the expected weight of the satisfied clauses is at least

$$\sum_{C|k_C \leq 2} w_C z_C \left(\frac{3}{2} q_1 + q_3 \cdot \alpha_0\right) + \sum_{C|k_C \geq 3} w_C z_C \cdot q_1 \left(1 - \frac{1}{2^k} + 1 - \left(1 - \frac{1}{k}\right)^k\right)$$

which can be verified to be at least $0.7554 \cdot \sum_C w_C z_C$.

The improved Feige-Goemans 0.931-approximation algorithm for MAX 2SAT. Feige and Goemans [63] show an improved performance ratio of about 0.931 for MAX 2SAT. For this, they strengthen the semidefinite relaxation (SDP) by adding to it the triangle inequalities:

$$X_{0i} + X_{0j} + X_{ij} \geq -1, \;\; X_{0i} - X_{0j} - X_{ij} \geq -1, \;\; -X_{0i} - X_{0j} + X_{ij} \geq -1 \tag{106}$$

for all $i, j \in \{1, \ldots, 2n\}$. Moreover, they replace the vectors $v_0, v_1, \ldots, v_n$ (obtained from the optimum solution to the strengthened semidefinite program) by a new set of vectors $v'_0, \ldots, v'_n$ obtained by applying some *rotation* to the v_i's. Then the assignment for the Boolean variables x_i are generated from the v'_i using as before the hyperplane rounding technique.

Let us explain how the vectors v'_i are generated from the v_i's. Let $f : [0, \pi] \to [0, \pi]$ be a continuous function such that $f(0) = 0$ and $f(\pi - \theta) = \pi - f(\theta)$. As before, θ_{ij} denotes the angle (v_i, v_j). The vector v_i is rotated in the plane spanned by v_0 and v_i until it forms an angle of $f(\theta_{0i})$ with v_0; the resulting vector is v'_i. If $v_i = v_0$ then $v'_i = v_i$. Moreover, let $v'_{n+i} = -v'_i$ for $i = 1, \ldots, n$. Let θ'_{ij} be the angle (v'_i, v'_j). Then $\theta'_{0i} = f(\theta_{0i})$ and Feige and Goemans [63] show the following equation permitting to express θ'_{ij} in terms of θ_{ij}:

$$\cos \theta'_{ij} = \cos \theta'_{0i} \cos \theta'_{0j} + \frac{\cos \theta_{ij} - \cos \theta_{0i} \cos \theta_{0j}}{\sin \theta_{0i} \sin \theta_{0j}} \sin \theta'_{0i} \sin \theta'_{0j}. \tag{107}$$

The probability that the clause $x_i \vee x_j$ is satisfied is now equal to

$$\text{prob}(v_0, v'_i, v'_j) = \frac{\theta'_{0i} + \theta'_{0j} + \theta'_{ij}}{2\pi}$$

while the contribution of this clause to the objective function of the semidefinite relaxation is

$$z_C \leq \frac{3 - \cos \theta_{0i} - \cos \theta_{0j} - \cos \theta_{ij}}{4}.$$

The performance ratio of the approximation algorithm using a rotation function f is, therefore, at least

$$\beta(f) := \min \frac{2}{\pi} \cdot \frac{\theta'_{01} + \theta'_{02} + \theta'_{12}}{3 - \cos \theta_{01} - \cos \theta_{02} - \cos \theta_{12}}$$

where the minimum is taken over all $\theta_{01}, \theta_{02}, \theta_{12} \in [0, \pi]$ for which $\cos \theta_{01}$, $\cos \theta_{02}$, $\cos \theta_{12}$ satisfy the triangle inequalities (106). Recall that $\theta'_{0i} = f(\theta_{0i})$ and relation (107) permits to express θ'_{12} in terms of θ_{01}, θ_{02} and θ_{12}.

Feige and Goemans [63] used a rotation function of the form

$$f_\lambda(\theta) = (1 - \lambda)\theta + \lambda \frac{\pi}{2}(1 - \cos \theta) \tag{108}$$

and, for the choice $\lambda = 0.806765$, they claim the lower bound 0.93109 for $\beta(f)$. Proving a correct evaluation of $\beta(f)$ is a non trivial task, since the minimization program defining $\beta(f)$ is too complicated to be handled analytically. Zwick [203] makes a detailed and rigorous analysis enabling him to prove a performance ratio of 0.931091 for MAX 2SAT.

The Matuura-Matsui 0.935-approximation algorithm for MAX 2SAT. Matuura and Matsui [141] design an approximation algorithm for MAX 2SAT with performance ratio 0.935. As in the Feige-Goemans algorithm, their starting point is to use the semidefinite relaxation (SDP') of MAX 2SAT obtained from (104) by adding the constraints (105) for the clauses of length 2 and the triangle inequalities (106); they fix v_0 to be equal to $(1,0,\ldots,0)^T$. Let $v_1,\ldots,v_n$ be the unit vectors obtained from an optimum solution to the program (SDP'). No rotation is applied to the vectors v_i as in the Feige-Goemans algorithm. The new ingredient in the algorithm of Matuura-Matsui consists of selecting the random hyperplane using a distribution function f on the sphere which is skewed towards v_0 and uniform in any direction orthogonal to v_0, instead of a uniform distribution.

Let $\mathcal{F}_n$ denote the set of functions $f : S^n \longrightarrow \mathbb{R}_+$ satisfying $\int_{S^n} f(v)dv = 1$, $f(v) = f(-v)$ for all $v \in S^n$, and $f(u) = f(v)$ for all $u, v \in S^n$ such that $u^T v_0 = v^T v_0$. Let $f \in \mathcal{F}_n$ and let the random unit vector r be now chosen according to the distribution function f. Then, $\text{prob}(v_i, v_j \mid f)$ denotes the probability that the clause $x_i \vee x_j$ is satisfied, i.e., as before, the probability that $\text{sign}(r^T v_0) \neq \text{sign}(r^T v_i)$ or $\text{sign}(r^T v_0) \neq \text{sign}(r^T v_j)$. Let P denote the linear subspace spanned by v_0, v_i, v_j and let $\hat{f}$ denote the distribution on S^2 obtained by projecting onto P; that is, $\hat{f}(v') := \int_{T(v')} f(v)dv$, where $T(v')$ is the set of all $v \in S^n$ whose projection on P is parallel to v'. Then the new approximation ratio of the algorithm is equal to

$$\alpha_{\hat{f}} := \min \frac{\text{prob}(v_i, v_j \mid \hat{f})}{1/4(3 - v_0^T v_i - v_0^T v_j - v_i^T v_j)}$$

where the minimum is taken over all $v_i, v_j \in S^2$ which together with $v_0 = (1,0,0)^T$ have their pairwise inner products satisfying the triangle inequalities (106).

The difficulty consists of constructing a distribution function $f \in \mathcal{F}_n$ for which $\alpha_{\hat{f}}$ is large. Matuura and Matsui [141] show the following. The function

$$g(v) := \cos^{1/1.3}(\theta) \text{ for all } v \in S^2 \text{ with } |v_0^T v| = \cos\theta, \tag{109}$$

is a distribution function on S^2 belonging to $\mathcal{F}_2$; it satisfies $\alpha_g \geq 0.935$ (this is proved numerically); and there exists $f \in \mathcal{F}_n$ for which $\hat{f} = g$.

The Karloff-Zwick $\frac{7}{8}$-approximation algorithm for MAX 3SAT. Karloff and Zwick [102] present an approximation algorithm for MAX 3SAT whose performance ratio they conjecture to be equal to $7/8 = 0.875$, thus the best possible since Håstad [89] proved the non existence of an approximation algorithm with performance ratio $> 7/8$ unless P=NP. Previous algorithms were using a reduction to the case of MAX 2SAT; for instance, Trevisan et al. [190] give a 0.801-approximation algorithm for MAX 3SAT using the Feige-Goemans 0.931 result for MAX 2SAT. Karloff and Zwick do not make such a reduction but consider instead the following direct semidefinite relaxation for MAX 3SAT:

$$\begin{array}{ll} \max & \sum_{i,j,k\in\{1,\ldots,2n\}} w_{ijk} z_{ijk} \\ \text{s.t.} & z_{ijk} \leq \text{relax}(v_0, v_i, v_j, v_k) \\ & v_i \cdot v_{n+i} = -1 \quad (i = 1,\ldots,n) \\ & v_0,\ldots,v_{2n} \in S^n,\ z_{ijk} \in \mathbb{R}, \end{array}$$

where z_{ijk} is a scalar attached to the clause $x_i \vee x_j \vee x_k$ and

$$\text{relax}(v_0, v_i, v_j, v_k) := \min(1 - \frac{(v_0 + v_i)^T(v_j + v_k)}{4}, 1 - \frac{(v_0 + v_j)^T(v_i + v_k)}{4}, 1 - \frac{(v_0 + v_k)^T(v_i + v_j)}{4}, 1).$$

Note indeed that when the v_i's are ± 1 scalars, then $\text{relax}(v_0, v_i, v_j, v_k)$ is equal to 0 precisely when $v_0 = v_i = v_j = v_k$ which corresponds to setting all variables x_i, x_j, x_k to 0 and thus to the clause $x_i \vee x_j \vee x_k$ not being satisfied.

Denote again by $\text{prob}(v_0, v_i, v_j, v_k)$ the probability that $x_i \vee x_j \vee x_k$ is satisfied and set

$$\text{ratio}(v_0, v_i, v_j, v_k) := \frac{\text{prob}(v_0, v_i, v_j, v_k)}{\text{relax}(v_0, v_i, v_j, v_k)}.$$

For a clause of length 1 or 2 (obtained by letting $j = k = 0$ or $k = 0$), it follows from the analysis of the GW algorithm that $\text{ratio}(v_0, v_i, v_j, v_k) \geq \alpha_0 > \frac{7}{8}$. For clauses of length 3, the analysis is technically much more involved and requires the computation of the volume of spherical tetrahedra as we now see.

Clearly, $\text{prob}(v_0, v_i, v_j, v_k)$ is equal to the probability that the random hyperplane H_r separates v_0 from at least one of v_i, v_j, v_k and thus to

$$1 - 2 \cdot \text{prob}(r^T v_h \geq 0 \ \forall h = 0, i, j, k).$$

We may assume without loss of generality that v_0, v_i, v_j, v_k lie in $\mathbb{R}^4$ and, since we are only interested in the inner products $r^T v_h$, we can replace r by its normalized projection on $\mathbb{R}^4$ which is then uniformely distributed on the sphere S^3. Define

$$T(v_0, v_i, v_j, v_k) := \{r \in S^3 \mid r^T v_h \geq 0 \ \forall h = 0, i, j, k\}.$$

Then, $\text{prob}(v_0, v_i, v_j, v_k) = 1 - 2 \cdot \frac{\text{vol}(T(v_0, v_i, v_j, v_k))}{\text{vol}(S^3)}$, where vol(.) denotes the 3-dimensional spherical volume. As $\text{vol}(S^3) = 2\pi^2$, we find that

$$\text{prob}(v_0, v_i, v_j, v_k) = 1 - 2 \cdot \frac{\text{vol}(T(v_0, v_i, v_j, v_k))}{\pi^2}.$$

When the vectors v_0, v_i, v_j, v_k are linearly independent, $T(v_0, v_i, v_j, v_k)$ is a spherical tetrahedron, whose vertices are the vectors $v'_0, v'_i, v'_j, v'_k \in S^3$ satisfying $v_h^T v'_h > 0$ for all h and $v_{h_1}^T v'_{h_2} = 0$ for all distinct h_1, h_2. That is,

$$T(v_0, v_i, v_j, v_k) = \{ \sum_{h=0,i,j,k} \alpha_h v'_h \mid \alpha_h \geq 0, \sum_h \alpha_h = 1\}.$$

Therefore, evaluating the quantity $\text{ratio}(v_0, v_i, v_j, v_k)$ and thus the performance ratio of the algorithm relies on proving certain inequalities about volumes of spherical tetrahedra.

Karloff and Zwick [102] show that $\text{prob}(v_0, v_i, v_j, v_k) \geq \frac{7}{8}$ whenever $\text{relax}(v_0, v_i, v_j, v_k) = 1$, which shows a performance ratio $\frac{7}{8}$ for satisfiable instances of MAX 3SAT. Their proof is computer assisted as it involves one computation carried out with Mathematica. Zwick [204] can prove the performance ratio $\frac{7}{8}$ for general MAX 3SAT. Although his proof is again computer assisted, it can however be considered as a rigorous proof since it is carried out using a new system called RealSearch, written by Zwick, which involves only interval arithmetic (instead of floating point arithmetic). We refer to Zwick's paper for an interesting presentation and discussion.

Further extensions. Karloff and Zwick [102] describe a procedure for constructing strong semidefinite relaxations for general constraint satisfaction problems and thus for MAX kSAT. Halperin and Zwick [87] study approximation algorithms for MAX 4SAT using the semidefinite relaxation provided by the Karloff-Zwick recipe. The analysis of the classic hyperplane rounding technique necessitates now the evaluation of the probability $\mathrm{prob}(v_0, \ldots, v_4)$ that a random hyperplane separates v_0 from at least one of $v_1, \ldots, v_4$. Luckily, using the inclusion/exclusion formula, this probability can be expressed in terms of the probabilities $\mathrm{prob}(v_i, v_j)$ and $\mathrm{prob}(v_i, v_j, v_k, v_\ell)$ that were considered above. In this way, Halperin and Zwick can show a performance ratio of 0.845173 for MAX 4SAT, thus below the target ratio of $\frac{7}{8}$. They study in detail a variety of other possible rounding strategies which enables them to obtain some improved performance ratio, like 0.8721.

6.7 Approximating the maximum directed cut problem

Given a directed graph $G = (V, A)$ and weights $w \in \mathbb{Q}_+^A$ associated to its arcs, the *maximum directed cut problem* asks for a directed cut $\delta^+(S)$ of maximum weight where, for $S \subseteq V$, the *directed cut* (or *dicut*) $\delta^+(S)$ is the set of arcs ij with $i \in S$ and $j \notin S$. This problem is NP-hard, since the max-cut problem in a undirected graph H reduces to the maximum dicut problem in the directed graph obtained by replacing each edge of H by two opposite arcs. Moreover, no approximation algorithm for the maximum dicut problem exists having a performance ratio $> \frac{12}{13}$ unless P=NP [89].

The simple random partition algorithm (which assigns each node to S independently with probability $\frac{1}{2}$) has a performance ratio $\frac{1}{4}$. Goemans and Williamson [77] show that their basic approximation algorithm for max-cut can be extended to the maximum dicut problem with performance ratio 0.79607. Feige and Goemans [63] prove an improved performance ratio of 0.859. These algorithms use the same ideas as the algorithms for MAX 2SAT presented in the same papers. Before presenting them, we mention a simple $\frac{1}{2}$-approximation algorithm of Halperin and Zwick [88] using a linear relaxation of the problem; this algorithm can in fact be turned into a purely combinatorial algorithm.

A $\frac{1}{2}$-approximation algorithm by Halperin and Zwick. Consider the following linear program:

$$\begin{array}{lll} \max & \sum_{ij \in A} w_{ij} z_{ij} & \\ \text{s.t.} & z_{ij} \leq x_i & (ij \in A) \\ & z_{ij} \leq 1 - x_j & (ij \in A) \\ & 0 \leq x_i \leq 1 & (i \in V). \end{array} \tag{110}$$

If we replace the linear constraint $0 \leq x \leq 1$ by the integer constraint $x \in \{0, 1\}^V$ then we obtain a formulation for the maximum dicut problem; the dicut $\delta^+(S)$ with $S = \{i \mid x_i = 1\}$ being an optimum dicut. Halperin and Zwick [88] show that the program (110) has a half-integer optimum solution. To see it, note first that (110) is equivalent to the program:

$$\begin{array}{lll} \max & \sum_{ij \in A} w_{ij} z_{ij} & \\ \text{s.t.} & z_{ij} + z_{jk} \leq 1 & (ij \in A,\ jk \in A) \\ & 0 \leq z_{ij} \leq 1 & (ij \in A). \end{array} \tag{111}$$

Indeed, if (z, x) is feasible for (110), then z is feasible for (111); conversely, if z is feasible for (111) then (z, x) is feasible for (110), where $x_i := \max_{ij \in A} z_{ij}$ if $\delta^+(i) \neq \emptyset$ and $x_i := 0$ otherwise. Now, the constraints in (111) define in fact the fractional stable set polytope of the line graph of G (whose nodes are the arcs, with two arcs being adjacent if they form a path in G). Since the vertices of the fractional stable set polytope are half-integral, it follows that (111) and thus (110) has a half-integral optimum solution (x, z). Then one construct a directed cut $\delta^+(S)$ by putting node $i \in V$ in S with probability x_i. The expected weight of $\delta^+(S)$ is at least $\frac{1}{2}w^T z$. Therefore, this gives a $\frac{1}{2}$-approximation algorithm. Moreover, this algorithm can be made purely combinatorial since a half-integral solution can be found using a bipartite matching algorithm (see [88]).

The Goemans-Williamson 0.796-approximation algorithm. One can alternatively model the maximum dicut problem in the following way. Given $v_0, v_1, \dots, v_n \in \{\pm 1\}$ and $S := \{i \in \{1, \dots, n\} \mid v_i = v_0\}$, the quantity

$$\frac{1}{4}(1 + v_0 \cdot v_i)(1 - v_0 \cdot v_j) = \frac{1}{4}(1 + v_0 \cdot v_i - v_0 \cdot v_j - v_i \cdot v_j)$$

is equal to 1 if $ij \in \delta^+(S)$ and to 0 otherwise. Therefore, the following program solves the maximum dicut problem:

$$\begin{array}{ll} \max & \sum_{ij \in A} w_{ij} \frac{1}{4}(1 + v_0 \cdot v_i - v_0 \cdot v_j - v_i \cdot v_j) \\ \text{s.t.} & v_0, v_1, \dots, v_n \in \{\pm 1\} \end{array} \tag{112}$$

Let (SDP) denote the relaxation of (112) obtained by replacing the condition $v_0, v_1, \dots, v_n \in \{\pm 1\}$ by the condition $v_0, v_1, \dots, v_n \in S^n$ and let z_{sdp} denote its optimum value. Goemans and Williamson propose the following analogue of their max-cut algorithm for solving the maximum dicut problem: Solve (SDP) and let $v_0, \dots, v_n$ be an optimum solution to it; select a random unit vector r and let $S := \{i \in \{1, \dots, n\} \mid \text{sign}(v_0 \cdot r) = \text{sign}(v_i \cdot r)\}$. Let θ_{ij} denote the angle (v_i, v_j). Then the expected weight $E(S)$ of the dicut $\delta^+(S)$ is equal to

$$E(S) = \sum_{ij \in A} w_{ij} \frac{1}{2\pi}(-\theta_{0i} + \theta_{0j} + \theta_{ij}).$$

In order to bound $\frac{E(S)}{z_{sdp}}$, one has to find lower bounds for the quantity

$$\frac{2}{\pi} \frac{-\theta_{0i} + \theta_{0j} + \theta_{ij}}{1 + \cos\theta_{0i} - \cos\theta_{0j} - \cos\theta_{ij}}.$$

Goemans and Williamson show the lower bound

$$\beta := \min_{0 \le \theta < \arccos(-1/3)} \frac{2}{\pi} \frac{2\pi - 3\theta}{1 + 3\cos\theta} > 0.79607$$

for it. Therefore, the above algorithm has performance ratio $\beta > 0.79607$.

The Feige-Goemans approximation algorithm. Feige and Goemans [63] propose an improved approximation algorithm for the maximum dicut problem analogue to their improved approximation algorithm for MAX 2SAT. Namely, strengthen the semidefinite program (SDP) by adding to it the triangle inequalities (106); replace the vectors $v_0, \dots, v_n$ obtained as optimum solution of the strengthened SDP program by a new set of vectors $v'_0, \dots, v'_n$ obtained by applying some rotation function to the v_i's; generate from the v'_i's the directed cut $\delta^+(S)$ where $S := \{i \in \{1, \dots, n\} \mid \text{sign}(v'_0 \cdot r) = \text{sign}(v'_i \cdot r)\}$. Thus one should now find lower bounds for the quantity

$$\frac{2}{\pi} \frac{-\theta'_{0i} + \theta'_{0j} + \theta'_{ij}}{1 + \cos\theta_{0i} - \cos\theta_{0j} - \cos\theta_{ij}}.$$

Using the rotation function f_λ from (108) with $\lambda = \frac{1}{2}$, Feige and Goemans claim a preformance ratio of 0.857. Zwick [203] makes a detailed analysis of their algorithm enabling him to show a performance ratio of 0.859643 (using an adequate rotation function).

The Matuura-Matsui 0.863**-approximation algorithm.** Matuura and Matsui [140] propose an approximation algorithm for the maximum directed cut problem with performance ratio 0.863. Analagously to their algorithm for MAX 2SAT presented in the previous subsection, it relies on solving the semidefinite relaxation strengthened by the triangle inequalities (106) and applying the random hyperplane rounding phase using a distribution on the sphere which is skewed towards v_0 and uniform in any direction orthogonal to v_0. As concrete choice, they propose to use the distribution function on S^2

$$g(v) = \cos^{1/1.8}(\theta) \quad \text{for all } v \in S^2 \text{ with } |v_0^T v| = \cos\theta \tag{113}$$

which can be realised as projection of a distribution on S^n and permits to show an approximation ratio of 0.863. (Compare (113) with the function g from (109) used for MAX 2SAT.)

7 Further Topics

7.1 Approximating polynomial programming using semidefinite programming

We come back in this section to the problem of approximating polynomial programs using semidefinite programming, which was already considered in Section 3.7. We present here the main ideas underlying this approach. They use results about representations of positive polynomials as sums of squares and moment sequences. Sums of squares will again be used in the next subsection for approximating the copositive cone. We then mention briefly some extensions to the general problem of testing whether a semi-algebraic set is empty.

Polynomial programs, sums of squares of polynomials, and moment sequences. Consider the following *polynomial programming problem:*

$$\min g(x) \;\text{ subject to }\; g_\ell(x) \geq 0 \;(\ell = 1, \ldots, m) \tag{114}$$

where g, g_ℓ are polynomials in $x = (x_1, \ldots, x_n)$. This is a very general problem which contains linear programming (when all polynomials have degree one) and 0/1 linear programming (since the integrality condition $x_i \in \{0, 1\}$ can be expressed as the polynomial equation: $x_i^2 - x_i = 0$). We mentioned in Section 3.7 that, under some technical assumption, the problem (114) can be approximated (getting arbitrarily close to its optimum) by the sequence of semidefinite programs (51). This result, due to Lasserre [115], relies on the fact that certain positive polynomials can be represented as sums of squares of polynomials. This idea of using sums of squares of polynomials for approximating polynomial programs has been introduced by Shor [183, 184, 185] and used by several other authors including Nesterov [151] and Parrilo [155, 156]; it seems to yield a more powerful method than other existing algebraic methods (see [157] for a comparison).

We would like to explain briefly here the main ideas underlying this approach. For simplicity, consider first the unconstrained problem:

$$p^* := \min g(x) \;\text{ subject to }\; x \in \mathbb{R}^n \tag{115}$$

where $g(x) = \sum_{\alpha \in S_{2d}} g_\alpha x^\alpha$ is a polynomial of even degree $2d$; here S_k denotes the set of sequences $\alpha \in \mathbb{Z}_+^n$ with $|\alpha| := \sum_{i=1}^n \alpha_i \leq k$ for any integer k. One can assume w.l.o.g. that $g(0) = g_0 = 0$. In what follows the polynomial $g(x)$ is identified with its sequence of coefficients $g = (g_\alpha)_{\alpha \in S_{2d}}$. Obviously, (115) can be rewritten as

$$p^* = \max \lambda \;\text{ subject to }\; g(x) - \lambda \geq 0 \;\forall x \in \mathbb{R}^n. \tag{116}$$

Testing whether a polynomial is nonnegative is a hard problem, since it contains the problem of testing whether a matrix is copositive (see the next subsection). Lower bounds for p^* can be obtained by considering sufficient conditions for the polynomial $g(x) - \lambda$ to be nonnegative on $\mathbb{R}^n$. An obvious such sufficient condition being that $g(x) - \lambda$ be a sum of squares of polynomials. Therefore,

$$p^* \geq \max \lambda \;\text{ subject to }\; g(x) - \lambda \text{ is a sum of squares.} \tag{117}$$

Testing whether a polynomial $p(x)$ is a a sum of squares amounts to testing feasibility of a semidefinite program (cf., e.g., [166]). Indeed, say $p(x)$ has degree $2d$, and let $z := (x^\alpha)_{\alpha \in S_d}$ be the vector consisting of all monomials of degree $\leq d$. Then one can easily verify that $p(x)$ is a sum of squares if and only if $p(x) = z^T X z$ (identical polynomials) for some positive semidefinite matrix X. For $\gamma \in S_{2d}$, set

$$B_\gamma := \sum_{\alpha, \beta \in S_d | \alpha + \beta = \gamma} E_{\alpha, \beta},$$

where $E_{\alpha,\beta}$ is the elementary matrix with all zero entries except entries 1 at positions (α,β) and (β,α).

Proposition 25. *A polynomial $p(x)$ of degree $2d$ is a sum of squares if and only if the following semidefinite program:*

$$X \succeq 0,\ \langle B_\gamma, X\rangle = p_\gamma\ (\gamma \in S_{2d}) \tag{118}$$

is feasible, where X is of order $\binom{n+d}{d}$ and with $\binom{n+2d}{2d}$ equations.

Proof. As $z^T X z = \sum_{\alpha,\beta\in S_d} X_{\alpha,\beta} x^{\alpha+\beta} = \sum_{\gamma\in S_{2d}} x^\gamma \left(\sum_{\substack{\alpha,\beta\in S_d\\ \alpha+\beta=\gamma}} X_{\alpha,\beta}\right) = \sum_{\gamma\in S_{2d}} x^\gamma \langle B_\gamma, X\rangle$, $p(x) = z^T X z$ for some $X \succeq 0$ (which is equivalent to $p(x)$ being a sum of squares) if and only if the system (118) is feasible. ▮

Note that the program (118) has a polynomial size for fixed n or d. Based on the result from Proposition 25, one can reformulate the lower bound for p^* from (117) as

$$p^* \geq \begin{array}{ll} \max & \lambda \\ \text{s.t.} & g(x) - \lambda \text{ is a sum of squares} \end{array} = \begin{array}{ll} \max & -\langle B_0, X\rangle \\ \text{s.t.} & \langle B_\gamma, X\rangle = g_\gamma\ (\gamma \in S_{2d} \setminus \{0\}). \end{array} \tag{119}$$

One can alternatively proceed in the following way for finding lower bounds for p^*. Obviously,

$$p^* = \min_\mu \int g(x) d\mu(x) \tag{120}$$

where the minimum is taken over all probability measures μ on $\mathbb{R}^n$. Define a sequence $y = (y_\alpha)_{\alpha\in S_{2d}}$ to be a *moment sequence* if $y_\alpha = \int x^\alpha d\mu(x)$ $(\alpha \in S_{2d})$ for some nonnegative measure μ on $\mathbb{R}^n$. Hence, (120) can be rewritten as

$$p^* = \min \sum_\alpha g_\alpha y_\alpha \ \text{ s.t. } y \text{ is a moment sequence and } y_0 = 1. \tag{121}$$

Lower bounds for p^* can be obtained by replacing the condition that y be a moment sequence by a necessary condition for it. An obvious such necessary condition is that the moment matrix $M_d^{\mathbb{Z}}(y) = (y_{\alpha+\beta})_{\alpha,\beta\in S_d}$ (recall (49)) be positive semidefinite. Thus we find the following lower bound for p^*:

$$p^* \geq \min g^T y \ \text{ subject to } \ M_d^{\mathbb{Z}}(y) \succeq 0 \text{ and } y_0 = 1. \tag{122}$$

Note that the constraint in (122) is precisely condition (51) (when there are no constraints $g_\ell(x) \geq 0$). Since $M_d^{\mathbb{Z}}(y) = B_0 y_0 + \sum_{\gamma\in S_{2d}\setminus\{0\}} B_\gamma y_\gamma$, the semidefinite programs in (122) and in (119) are in fact dual of each other, which reflects the duality existing between the theories of nonnegative polynomials and of moment sequences.

The lower bound from (119) is equal to p^* if $g(x) - p^*$ is a sum of squares; this holds for $n = 1$ but not in general if $n \geq 2$. In general one can estimate p^* asymptotically by a sequence of SDP's analogue to (119) if one assumes that an upper bound R is known a priori on the norm of a global minimizer x of $g(x)$, in which case

$$p^* = \min g(x) \text{ subject to } g_1(x) := R - \sum_{i=1}^n x_i^2 \geq 0.$$

Indeed, one can then use a result of Putinar [167] (quoted in Theorem 26 below) and conclude that, for any $\epsilon > 0$, the polynomial $g(x) - p^* + \epsilon$ is *positive* on $F := \{x \mid g_1(x) \geq 0\}$ and thus can be decomposed as $p(x) + p_1(x)g_1(x)$ for some polynomials $p(x)$ and $p_1(x)$ that are sums of squares. Testing for the existence of such decomposition can be expressed as a SDP program analogue to (119). Its dual (analogue to (122)) reads:

$$p_t^* := \min \ g^T y \ \text{ subject to } M_t(y) \geq 0, \ M_{t-1}(g_1 * y) \succeq 0, \ y_0 = 1.$$

Putinar's result permits to show the asymptotic convergence of p_t^* to p^* when t goes to infinity.

Theorem 26. [167] *Let $g_1, \ldots, g_m$ be polynomials and set $F := \{x \in \mathbb{R}^n \mid g_1(x) \geq 0, \ldots, g_m(x) \geq 0\}$. Assume that F is compact and that there exists a polynomial u satisfying (i) the set $\{x \in \mathbb{R}^n \mid u(x) \geq 0\}$ is compact and (ii) u can be decomposed as $u_0 + \sum_{\ell=1}^m u_\ell g_\ell$ for some polynomials $u_0, \ldots, u_m$ that are sums of squares. Then every polynomial $p(x)$ which is positive on F can be decomposed as $p = p_0 + \sum_{\ell=1}^m p_\ell g_\ell$ for some polynomials $p_0, \ldots, p_m$ that are sums of squares.*

The above reasoning extends to the general program (114) if the assumption of Theorem 26 holds. This is the case, e.g., if the set $\{x \mid g_\ell(x) \geq 0\}$ is compact for one of the polynomials defining F. Then, Putinar's result permits to claim that, for any $\epsilon > 0$, the polynomial $g(x) - p^* + \epsilon$ can be decomposed as $p(x) + \sum_{\ell=1}^m p_\ell(x) g_\ell(x)$ for some polynomials $p(x)$, $p_\ell(x)$ that are sums of squares. Based on this, on can derive the asymptotic convergence to p^* of the minimum of $g^T y$ taken over all y satisfying (51) when t goes to ∞. In the 0/1 case, when the constraints $x_i^2 - x_i = 0$ $(i = 1, \ldots, n)$ are part of the system defining F, there is in fact finite convergence in n steps [116] (see Section 3).

Semidefinite programming and the Positivstellensatz. Consider the following system:

$$\begin{array}{l} f_j(x) \geq 0 \ (j = 1, \ldots, s) \\ g_k(x) \neq 0 \ (k = 1, \ldots, t) \\ h_\ell(x) = 0 \ (\ell = 1, \ldots, u) \end{array} \tag{123}$$

where all f_j, g_k, h_ℓ are polynomials in the real variable $x = (x_1, \ldots, x_n)$. The complexity of the problem of testing feasibility of this system has been the object of intensive research. Tarski [1951] showed that this problem is decidable and since then a number of other algorithms have been proposed, in particular, by Renegar [173] and Basu et al. [28].

We saw in Proposition 25 that testing whether a polynomial is a sum of squares can be formulated as a semidefinite program. Parrilo [155] showed that the general problem of testing infeasibility of the system (123) can also be formulated as a semidefinite programming problem (of very large size). This is based on the following result of real algebraic geometry, known as the 'Positivstellensatz'. The Positivstellensatz asserts that for a system of polynomial (in)equalities, either there is a solution in $\mathbb{R}^n$, or there is a polynomial identity giving a certificate that no real solution exists. This gives therefore a common generalization of Hilbert's 'Nullstellensatz' (in the complex case) and Farkas' lemma (for linear systems).

Theorem 27. ([187], [33]) *The system* (123) *is infeasible if and only if there exist polynomials f, g, h*

of the form

$$\begin{aligned} f(x) &= \sum_{S\subseteq\{1,\dots,s\}} p_S \left(\prod_{j\in S} f_j\right) \quad \textit{where all } p_S \textit{ are sums of squares} \\ g(x) &= \prod_{k\in K} g_k \quad \textit{where } K \subseteq \{1,\dots,t\} \\ h(x) &= \sum_{\ell=1}^{u} q_\ell h_\ell \quad \textit{where all } q_\ell \textit{ are polynomials} \end{aligned}$$

satisfying the equality $f + g^2 + h = 0$.

Bounds are known a priori for the degrees of the polynomials in the Positivstellensatz which make it possible to test infeasibility of the system (123) via semidefinite programming. However, these bounds are very large (triply exponential in n). Practically, one can use semidefinite programming for searching for infeasibility certificates of bounded degree.

7.2 Approximating combinatorial problems using copositive programming

We have seen throughout this chapter how semidefinite programming can be used for approximating combinatorial optimization problems. The idea of using the copositive cone and its dual, the cone of completely positive matrices, instead of the positive semidefinite cone has also been considered; cf., e.g., [35], [168]. We present below some results of de Klerk and Pasechnik [110] showing how the stability number of a graph can be computed using copositive relaxations. Extensions to standard quadratic optimization problems are given by Bomze and de Klerk [36].

Let us first recall some definitions. A symmetric matrix M of order n is *copositive* if $x^T M x \geq 0$ for all $x \in \mathbb{R}^n_+$ and M is *completely positive* if $M = \sum_{i=1}^k u_i u_i^T$ for some nonegative vectors $u_1, \dots, u_k$. Let $\mathcal{C}_n$ denote the set of symmetric copositive matrices of order n; its dual cone $\mathcal{C}_n^*$ is the set of completely positive matrices. Hence,

$$\mathcal{C}_n^* \subseteq \mathrm{PSD}_n = \mathrm{PSD}_n^* \subseteq \mathcal{C}_n.$$

Testing whether a matrix M is copositive is a co-NP-complete problem [147].

Let $G = (V, E)$ ($V = \{1, \dots, n\}$) be a graph and consider its theta number $\vartheta(G)$, defined by

$$\vartheta(G) = \max \langle J, X\rangle \text{ s.t. } X_{ij} = 0 \ (ij \in E), \mathrm{Tr}(X) = 1, \ X \succeq 0 \tag{124}$$

(same as definition (53)). Then, $\vartheta(G)$ is an upper bound for the stability number of G, since for any stable set S in G, the matrix $X_S := \frac{1}{|S|}\chi^S(\chi^S)^T$ is feasible for the semidefinite program (124). Note that X_S is in fact completely positive. Therefore, one can define a tighter upper bound for $\alpha(G)$ by replacing in (124) the condition $X \succeq 0$ by the condition $X \in \mathcal{C}_n^*$. Letting A denote the adjacency matrix of G, we obtain:

$$\begin{array}{llllll} \alpha(G) \leq & \max & \langle J, X\rangle & \leq & \min & \lambda \\ & \text{s.t.} & \mathrm{Tr}X = 1 & & \text{s.t.} & \lambda I + yA - J \in \mathcal{C}_n \\ & & X_{ij} = 0 \ (ij \in E) & & & \lambda, y \in \mathbb{R} \\ & & X \in \mathcal{C}_n^* & & & \end{array} \tag{125}$$

where the right most program is obtained from the left most one using cone-LP duality. Using the following formulation for $\alpha(G)$ due to Motzkin and Straus [146]:

$$\frac{1}{\alpha(G)} = \min \ x^T(A+I)x \ \text{ subject to } \ x \geq 0 \text{ and } \sum_{i=1}^n x_i = 1,$$

one finds that the matrix $\alpha(G)(I+A)-J$ is copositive. This implies that the optimum value of the right most program in (125) is at most $\alpha(G)$. Therefore, equality holds throughout in (125). This shows again that copositive programming is not tractable.

Parrilo [155] proposes to approximate the copositive cone using sums of squares of polynomials. For this, note that a matrix M is copositive if and only if the polynomial

$$g_M(x) := \sum_{i,j=1}^{n} M_{ij} x_i^2 x_j^2$$

is nonnegative on $\mathbb{R}^n$. Therefore, an obvious sufficient condition for M to be copositive is that $g_M(x)$ be a sum of squares or, more generally, that the polynomial $g_M(x)\left(\sum_{i=1}^{n} x_i^2\right)^r$ be a sum of squares for some integer $r \geq 0$. A theorem of Polya asserts that, conversely, if M is strictly copositive (i.e., $x^T M x > 0$ for all $x \in \mathbb{R}^n_+ \setminus \{0\}$), then $g_M(x)\left(\sum_{i=1}^{n} x_i^2\right)^r$ is a sum of squares for some r. Powers and Reznick [165] give some upper bound for this integer r (depending only on M).

Let $\mathcal{K}_n^r$ denote the set of symmetric matrices M of order n for which $g_M(x)\left(\sum_{i=1}^{n} x_i^2\right)^r$ is a sum of squares. Thus

$$\mathrm{PSD}_n \subseteq \mathcal{K}_n^0 \subseteq \ldots \subseteq \mathcal{K}_n^r \subseteq \mathcal{C}_n.$$

We saw in the preceding subsection that testing whether a polynomial is a sum of squares can be solved via the semidefinite program (118). Therefore one can test membership in $\mathcal{K}_n^r$ via semidefinite programming. For instance, Parrilo [155] shows that

$$M \in \mathcal{K}_n^0 \Longleftrightarrow M = P + N \text{ for some } P \succeq 0,\ N \geq 0.$$

Moreover, $M \in \mathcal{K}_n^1$ if and only if the following system:

$$\begin{array}{rll} M - X^{(i)} & \succeq 0 & (i = 1, \ldots, n) \\ X_{ii}^{(i)} & = 0 & (i = 1, \ldots, n) \\ X_{ii}^{(j)} + 2X_{ij}^{(i)} & = 0 & (i \neq j = 1, \ldots, n) \\ X_{jk}^{(i)} + X_{ik}^{(j)} + X_{ij}^{(k)} & \geq 0 & (1 \leq i < j < k \leq n) \end{array}$$

has a solution, where $X^{(1)}, \ldots, X^{(n)}$ are symmetric $n \times n$ matrices ([155] and [36]).

Replacing in (125) the condition $\lambda I + yA - J \in \mathcal{C}_n$ by the condition $\lambda I + yA - J \in \mathcal{K}_n^r$, one can define the parameter

$$\vartheta^r(G) := \min \lambda \ \text{ subject to } \ \lambda I + yA - J \in \mathcal{K}_n^r.$$

Using the bound of Powers and Reznick [165], de Klerk and Pasechnik [110] show that

$$\alpha(G) = \lfloor \vartheta^r(G) \rfloor \quad \text{if } r \geq \alpha^2(G).$$

The same conclusion holds if we replace $\mathcal{K}_n^r$ by the cone $\mathcal{C}_n^r$ consisting of the matrices M for which $g_M(x)\left(\sum_{i=1}^{n} x_i^2\right)^r$ has only nonnegative coefficients. Bomze and de Klerk [36] characterized the cone $\mathcal{C}_n^r$ as the set of matrices M satisfying: $x^T M x - x^T \mathrm{diag}(M) \geq 0$ for all $x \in \mathbb{Z}^n_+$ with $\sum_{i=1}^n x_i = r+2$. It is also shown in [110] that $\vartheta^0(G) = \vartheta'(G)$, the Schrijver parameter from (60); $\vartheta^1(G) = \alpha(G)$ if G is an odd circuit, an odd wheel or their complement, or if $\alpha(G) = 2$. It is conjectured in [110] that $\vartheta^{\alpha(G)-1}(G) = \alpha(G)$.

8 Semidefinite Programming and the Quadratic Assignment Problem

Quadratic problems in binary variables are the prime source for semidefinite models in combinatorial optimization. The simplest form, unconstrained quadratic programming in binary variables, corresponds to Max-Cut, and was described in detail in Section 5.

Assuming that the binary variables are the elements of a permutation matrix leads to the *Quadratic Assignment Problem (QAP)*. Formally, QAP consists in minimizing

$$\operatorname{Tr}(AXB + C)X^T \tag{126}$$

over all permutation matrices X. One usually assumes that A and B are symmetric matrices of order n, while the linear term C is an arbitrary matrix of order n. There are many applications of this model problem, for instance in location theory. We refer to the recent monograph [40] for a description of published applications of QAP in Operations Research and combinatorial optimization.

The cost function (126) is quadratic in the matrix variable X. To rewrite this we use the vec-operator and (9). This leads to

$$\operatorname{Tr}AXBX^T = \langle vec(X), vec(AXB)\rangle = x^T(B \otimes A)x, \tag{127}$$

because B is assumed to be symmetric. We can therefore express QAP equivalently as

$$\min\{x^T(B \otimes A)x + c^Tx : x = vec(X), X \text{ permutation matrix}\}.$$

Here, $c = vec(C)$. To derive semidefinite relaxations of QAP we follow the generic pattern and linearize by introducing a new matrix variable for xx^T, leading to the study of

$$P = \operatorname{conv}\{xx^T : x = vec(X), X \text{ permutation matrix}\}.$$

In section 3, we observed that any $Y \in P$ must satisfy the semidefiniteness condition (20), which in our present notation amounts to

$$Z = \begin{pmatrix} 1 & z^T \\ z & Y \end{pmatrix} \succeq 0, \ \operatorname{diag}(Y) = z.$$

The first question is to identify the smallest subcone of semidefinite matrices that contains P.

We use the following parametrization of matrices having row and column sums equal to e, the vector of all ones, see [84].

Lemma 28. [84] *Let V be an $n \times (n-1)$ matrix with $V^Te = 0$ and* $\operatorname{rank}(V) = n - 1$. *Then*

$$\mathcal{E} := \{X \in \mathbb{R}^{n\times n} : Xe = X^Te = e\} = \{\frac{1}{n}ee^T + VMV^T : M \in \mathbb{R}^{(n-1)\times(n-1)}\} =: \mathcal{E}'.$$

Proof. Let $Z = \frac{1}{n}ee^T + VMV^T \in \mathcal{E}'$. Then $Ze = Z^Te = e$, because $V^Te = 0$, hence $Z \in \mathcal{E}$. To see the other inclusion, let $V = QR$ be the QR-decomposition of V, i.e. $Q^TQ = I, QQ^T = I - \frac{1}{n}ee^T$ and $\operatorname{rank}(R) = n - 1$. Let $X \in \mathcal{E}$ and set $M := R^{-1}Q^TXQ(R^{-1})^T$. Then $\frac{1}{n}ee^T + VMV^T = X \in \mathcal{E}'$. ∎

We use this parametrization and define

$$W := \left(\frac{1}{n}e \otimes e, V \otimes V\right).$$

V can be any basis of $e^{\perp}$, as in the previous lemma. We can now describe the smallest subcone containing P.

Lemma 29. *Let $Y \in P$. Then there exists a symmetric matrix R of order $(n-1)^2+1$, indexed from 0 to $(n-1)^2$, such that*

$$R \succeq 0, \; r_{00} = 1, \; Y = WRW^T.$$

Proof. (See also [201].) We first look at the extreme points of P, so let X be a permutation matrix. Thus we can write X as $X = \frac{1}{n}ee^T + VMV^T$, for some matrix M. Let $m = vec(M)$. Then, using (9),

$$x = vec(X) = \frac{1}{n}e \otimes e + (V \otimes V)m = Wz,$$

with $z = \begin{pmatrix} 1 \\ m \end{pmatrix}$. Now $xx^T = Wzz^TW^T = WRW^T$, with $r_{00} = 1, \; R \succeq 0$. The same holds for convex conbinations formed from several permutation matrices. ∎

To see that the set

$$\hat{P} := \{Y : \exists R \text{ such that } Y = WRW^T, \; z = \operatorname{diag}(Y), \; \begin{pmatrix} 1 & z^T \\ z & Y \end{pmatrix} \succeq 0\} \tag{128}$$

is indeed the smallest subcone of positive semidefinite matrices containing P, it is sufficient to provide a positive definite matrix $\hat{R}$, such that $W\hat{R}W^T \in P$. In [201] it is shown that

$$\hat{R} = \begin{pmatrix} 1 & 0 \\ 0 & \frac{1}{n^2(n-1)}(nI_{n-1} - E_{n-1}) \otimes (nI_{n-1} - E_{n-1}) \end{pmatrix} \succ 0$$

gives

$$W\hat{R}W^T = \frac{1}{n!}\sum_{X \in \Pi}(xx^T),$$

the barycenter of P. Here $V = \begin{pmatrix} I_{n-1} \\ -e_{n-1}^T \end{pmatrix}$ has to be used in the definition of W.

Eliminating Y leaves the matrix variable R and n^2+1 equality constraints, fixing the first row equal to the main diagonal, and setting the first element equal to 1.

Thus we arrive at the following basic SDP relaxation of QAP:

$$(QAP_{R1}) \quad \min \operatorname{Tr}(B \otimes A + \operatorname{Diag}(c))Y \text{ such that } Y = WRW^T \in \hat{P}, \; r_{00} = 1. \tag{129}$$

It is instructive to look at $W\hat{R}W^T$ for small values of n. For $n = 3$ we get

$$W\hat{R}W^T = \frac{1}{6}\left(\begin{array}{ccc|ccc|ccc} 2 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 2 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 2 & 1 & 1 & 0 & 1 & 1 & 0 \\ \hline 0 & 1 & 1 & 2 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 2 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 2 & 1 & 1 & 0 \\ \hline 0 & 1 & 1 & 0 & 1 & 1 & 2 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 & 2 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 2 \end{array}\right).$$

The zero pattern in this matrix is not incidental. In fact, any $X \in P$ will have entries equal 0 at positions corresponding to $x_{ij}x_{ik}$ and $x_{ji}x_{ki}$ for $j \neq k$. This corresponds to the off-diagonal elements of the main diagonal blocks, and the main-diagonal elements of the off diagonal blocks. To express these constraints, we introduce some more notation, and index the elements of matrices in P alternatively by $P = (p_{(i,j),(k,l)})$ for i, j, k, l between 1 and n.

Hence we can strengthen the above relaxation by asking that

$$y_{rs} = 0 \text{ for } r = (i,j),\ \ s = (i,k),\ \text{ or } r = (j,i),\ \ s = (k,j),\ \ j \neq k.$$

We collect all these equations in the constraint $G(Y) = 0$. Adding it to (129) results in a stronger relaxation. In [201] this model is called the 'Gangster model'. Aside from $n^2 + 1$ equality constraints from the basic model, we have $O(n^3)$ equations in this extended model. This amounts to serious computational work, but results in a very strong lower bound for QAP.

$$(QAP_{R2}) \quad \min \operatorname{Tr}(B \otimes A + \operatorname{Diag}(c))Y \text{ such that } Y = WRW^T \in \hat{P},\ r_{00} = 1,\ G(Y) = 0. \tag{130}$$

Finally, one can include the constraints $y_{rs} \geq 0$ for all r, s, leading to

$$(QAP_{R3}) \quad \min \operatorname{Tr}(B \otimes A + \operatorname{Diag}(c))Y \text{ such that } Y = WTRW^T \in \hat{P},\ r_{00} = 1,\ G(Y) = 0,\ Y \geq 0. \tag{131}$$

The resulting SDP has $O(n^4)$ constraints and can not be solved in a straightforward way by interior point methods for problems of interesting size ($n \geq 15$).

The Anstreicher-Brixius bound. Anstreicher et al. [12, 13] have recently achieved a breakthrough in solving several instances of QAP which could not be solved by previous methods. The size of these instances ranges from $n = 20$ to $n = 36$. The key to this breakthrough lies in the use of a bound for QAP that is both 'fast' to compute, and gives 'good' approximations to the exact value of QAP. This bounding procedure combines orthogonal, semidefinite and convex quadratic relaxations in a nontrivial way, starting from the Hoffman-Wielandt inequality, Theorem 5.

A simple way to derive this bound goes as follows. We use the parametrization

$$X = \frac{1}{n}ee^T + VYV^T \tag{132}$$

from Lemma 28, and assume in addition that $V^TV = I_{n-1}$. Substituting this into the cost of function of QAP results in

$$\operatorname{Tr}(AXB + C)X^T = \operatorname{Tr}\hat{A}Y\hat{B}Y^T + \operatorname{Tr}(\hat{C} + \frac{2}{n}V^TAee^TBV)Y^T + \frac{1}{n^2}s(A)s(B) + \frac{1}{n}s(C), \tag{133}$$

where $\hat{A} = V^T A V, \hat{B} = V^T B V, \hat{C} = V^T C V$ and $s(M) := e^T M e = \sum_{ij} m_{ij}$. The condition $V^T V = I$ implies that X in (132) is orthogonal if and only if Y is. Hadley et al. [84] use this to bound the quadratic term in Y by the minimal scalar product of the eigenvalues of $\hat{A}$ and $\hat{B}$, see Theorem 5. Anstreicher and Brixius [12] use this observation as a starting point and observe that for any symmetric matrix $\hat{S}$, and any orthogonal Y, one has

$$0 = \mathrm{Tr}\hat{S}(I - YY^T) = \mathrm{Tr}\hat{S} - \mathrm{Tr}\hat{S}YIY^T = \mathrm{Tr}\hat{S} - \mathrm{Tr}(I \otimes \hat{S})(yy^T).$$

This results in the following identity, true for any orthogonal Y and any symmetric $\hat{S}, \hat{T}$:

$$\mathrm{Tr}\hat{A}Y\hat{B}Y^T = \mathrm{Tr}(\hat{S} + \hat{T}) + \mathrm{Tr}(\hat{B} \otimes \hat{A} - I \otimes \hat{S} - \hat{T} \otimes I)(yy^T). \tag{134}$$

We use $\hat{Q} = \hat{B} \otimes \hat{A} - I \otimes \hat{S} - \hat{T} \otimes I, \hat{D} = \hat{C} + \frac{2}{n} V^T A e e^T B V$ and substitute this into (133) to get

$$\mathrm{Tr}(AXB + C)X^T = \mathrm{Tr}(\hat{S} + \hat{T}) + y^T \hat{Q} y + \hat{d}^T y + \frac{1}{n^2} s(A)s(B) + \frac{1}{n} s(C), \tag{135}$$

This relation is true for any orthogonal X and Y related by (132) and symmetric $\hat{S}, \hat{T}$. It is useful to express the parts in (135) containing Y by the original matrix X. To do this we use the following identity:

$$0 = \mathrm{Tr}\hat{S}(I - V^T V) = \mathrm{Tr}\hat{S}(I - V^T X X^T V) = \mathrm{Tr}\hat{S} - \mathrm{Tr}(V\hat{S}V^T)XIX^T = \mathrm{Tr}\hat{S} - \mathrm{Tr}(I \otimes V\hat{S}V^T)(xx^T).$$

Hence, for any orthogonal X, and any symmetric $\hat{S}, \hat{T}$ we also have

$$\mathrm{Tr}(AXB + C)X^T = \mathrm{Tr}(\hat{S} + \hat{T}) + x^T Q x + c^T x. \tag{136}$$

Here $Q = B \otimes A - I \otimes (V\hat{S}V^T) - (V\hat{T}V^T) \otimes I$. Comparing (135) and (136) we note that

$$y^T \hat{Q} y + \hat{d}^T y + \frac{1}{n^2} s(A)s(B) + \frac{1}{n} s(C) = x^T Q x + c^T x.$$

It should be observed that Q and $\hat{Q}$ above depend on the specific choice of $\hat{S}, \hat{T}$. Anstreicher and Brixius use the optimal solution $\hat{S}, \hat{T}$ from Theorem 6 and observe that dual feasibility yields $\hat{Q} \succeq 0$. Therefore the above problem is a convex quadratic programming problem. We denote its optimal solution as the Anstreicher-Brixius bound ABB(A,B,C).

$$ABB(A, B, C) := \mathrm{Tr}(\hat{S} + \hat{T}) + \min\{x^T Q x + c^T x : x = vec(X), X \text{ doubly stochastic}\}.$$

The interesting observation here is that $\hat{S}, \hat{T}$ are obtained as a by-product of the Hoffman-Wielandt inequality, and that the resulting matrix Q is positive semidefinite over the set of doubly stochastic matrices (as a consequence of Theorem 6). These facts imply that the Anstreicher-Brixius bound is tractable.

To give a flavour of the quality of these bounds, we provide the following computational results on standard test sets from Nugent et al. [153]. These data sets have the following characteristics. The linear term C is equal to 0. The matrix B represents the rectilinear cell distance of a rectangular array of cells, hence there is some symmetry in these data. In case of $n = 12$, the resulting rectangular cell array has the following form:

1	2	3	4
5	6	7	8
9	10	11	12

We observe that the distance matrix B would not change, if the following cell array would have been used:

4	3	2	1
8	7	6	5
12	11	10	9

Mathematically speaking, there exist several permutation matrices X, such that $B = XBX^T$. Exploiting all these symmetries, it is sufficient to consider only the subproblems where the cells $1, 2, 5, 6$ are assigned to some fixed location, say 1. All other permutations can be obtained by exploiting the automorphisms inherent in B.

We denote these subproblems by nug12.1, nug12.2, nug12.5, nug12.6 in Table 1. The instance $n = 15$ has a distance matrix B corresponding to a 5×3 rectangular grid, leading to subproblems nug15.1, nug15.2, nug15.3, nug15.6, nug15.7, nug15.8. The optimal values for these instances are contained in the column labeled 'exact'. These values can be computed routinely for $n \approx 15$. The biggest instance $n = 30$ was only recently solved to optimality, see [13]. The computational results for QAP_{R3} are from the forthcoming dissertation [186]. It is computationally infeasible to solve this relaxation by interior points. Sotirov [186] uses the bundle method to get approximate solutions of QAP_{R3}. Hence the values are only lower estimates of the true bound. The values of QAP_{R2} were obtained by Sotirov and Wolkowicz [2] by making use of the NEOS distributed computing system. The bounds are obtained using interior point methods. The computational effort to get these values is prohibitively big. A more practical approach consists in using bundle methods to bargain computational efficiency against a slight decrease in the quality of the bound. Finally, the values of the Anstreicher-Brixius bound ABB are from [12].

These results indicate that the SDP models in combination with bundle methods may open the way to improved Branch and Bound approaches to solve larger QAP instances.

9 Epilogue: Semidefinite Programming and Algebraic Connectivity

An implicit message of all the preceeding sections is that semidefinite programming relaxations have a high potential to significantly improve on purely polyhedral relaxations. This may give the wrong impression that semidefinite programming is a universal remedy to improve upon linear relaxations. This is in principle true, if we assume that some sort of semidefiniteness constraint is *added* to the polyhedral model.

If a model based on semidefinite programming is used *instead* of a linear model, it need not be true that the semidefinite model dominates the linear one. We conclude with an illustration of this perhaps not quite intuitive statement.

We consider the Traveling Salesman Problem (TSP), i.e. the problem of finding a shortest Hamiltonian cycle in an edge weighted graph. This problem is well known to be NP-hard, and has stimulated research since the late 1950's.

We need to recall some notation from graph theory. For an edge weighted graph, given by its weighted adjacency matrix X, with $X \geq 0$, $\operatorname{diag}(X) = 0$ (setting to 0 the entries corresponding to nonedges), we consider vertex partitions $(S, V \setminus S)$ of its vertex set V and define

$$X(S, V \setminus S) := \sum_{i \in S, j \notin S} x_{ij}$$

[2] personal communication, 2001

problem	exact	QAP_{R2}	QAP_{R3}	ABB
nug12	578	529.3	552.1	482
nug12.1	586	550.7	573.6	-
nug12.2	586	550.6	571.3	-
nug12.5	578	551.8	572.2	-
nug12.6	600	555.8	578.8	-
nug15	1150	1070.5	1106.1	996
nug15.1	1150	1103.4	1131.6	-
nug15.2	1168	1116.3	1147.8	-
nug15.3	1164	1120.9	1148.4	-
nug15.6	1166	1113.6	1144.9	-
nug15.7	1182	1130.3	1161.9	-
nug15.8	1184	1134.1	1162.2	-
nug20	2570	2385.6	2441.9	2254
nug30	6124	5695.4	5803.2	5365

Table 1: Semidefinite relaxations and optimal value for some instances from the Nugent collection of test data. The column labeled QAp_{R3} gives lower estimates of the bound computed by the bundle method.

to be the weight of the cut, given by S. The *edge connectivity* $\mu(X)$ of X is defined as

$$\mu(X) := \min\{X(S, V \setminus S) : S \subseteq V,\ 1 \leq |S| \leq |V| - 1\}.$$

The polyhedral approach to TSP is based on approximating the convex hull of all Hamiltonian cycles by considering all two-edge connected graphs. Formally, this amounts to optimizing over the following set:

$$\{X : 0 \leq x_{ij} \leq 1,\ \operatorname{diag}(X) = 0,\ Xe = 2e,\ \mu(X) = 2\}. \tag{137}$$

Even though there are $O(2^n)$ linear constraints defining this (polyhedral) set, it is possible to optimize over it in polynomial time, by using the ellipsoid method (because the separation problem amounts to a minimum capacity cut problem, which can thus be solved in polynomial time). It is also interesting to note that no combinatorial algorithm of provably polynomial running time exists for optimizing a linear function over this set.

Recently, Cvetcovic et al. [51] have proposed a model where 2-edge connectivity is replaced by the algebraic connectivity, leading to an SDP relaxation.

Fiedler [68] introduces the algebraic connectivity of a graph, given by its weighted adjacency matrix $X \geq 0, \operatorname{diag}(X) = 0$, as follows. Let $L(X) := D - X$ be the Laplacian matrix corresponding to X, where $D := Diag(Xe)$, the diagonal matrix having the row sums of X on its main diagonal. Since $De = Xe$, it is clear that 0 is an eigenvalue of $L(X)$ corresponding to the eigenvector e. Moreover $X \geq 0$ implies by the Gersgorin disk theorem, that all eigenvalues of $L(X)$ are nonnegative, i.e., $L(X)$ is positive semidefinite in this case. Fiedler observed that the second smallest eigenvalue $\lambda_2(L(X)) = \min_{\|u\|=1, u^Te=0} u^T L(X) u$ is equal to 0 if and only if X is the adjacency matrix of a disconnected graph, otherwise $\lambda_2(L(X)) > 0$. Note also that $\lambda_2(L(X))$ is concave in X. Fiedler therefore denotes $\alpha(X) := \lambda_2(L(X))$ as the algebraic connectivity of the graph, given by the adjacency matrix X. It is not difficult to calculate $\alpha(C_n)$, the algebraic connectivity of a cycle on n nodes,

$$\alpha(C_n) = 2(1 - \cos(\frac{2\pi}{n})) =: h_n$$

The concavity of $\alpha(X)$ therefore implies that

$$\alpha(X) \geq h_n$$

for any convex combination X of Hamiltonian cycles. We also note that the Taylor expansion of $\cos(x)$ gives $h_n \leq \frac{4\pi^2}{n^2}$. Cvetcovic et al. [51] propose to replace the polyhedral constraints $\mu(X) \geq 2$ by the nonlinear condition $\alpha(X) \geq h_n$, which can easily be shown to be equivalent to the semidefiniteness constraint

$$L(X) + ee^T - h_n I \succeq 0$$

on X. Replacing edge connectivity by algebraic connectivity in (137) leads to optimizing over

$$\{X : 0 \leq x_{ij} \leq 1,\ \operatorname{diag}(X) = 0,\ Xe = 2e,\ L(X) + ee^T - h_n I \succeq 0\}. \tag{138}$$

This looks like a reasonable bargain, as we replace $O(2^n)$ linear constraints by a single semidefiniteness constraint. The crucial question of course is whether we can say anything about the relative strength of the two relaxations. Since $L(X) + ee^T \succeq 0$ it is clear that

$$\lambda_{\min}(L(X) + ee^T - h_n I) \geq -h_n \geq -\frac{4\pi^2}{n^2}.$$

Therefore the semidefiniteness constraint in (138) is nearly satisfied for any $X \geq 0$ as the dimension increases. We can say even more. Any matrix X feasible for (137) satisfies $\alpha(X) \geq h_n$, see [67] and the handbook [194], chapter 12 for further details. In other words, the simple semidefinite relaxation given by (138) is dominated by the polyhedral edge connectivity model (137).

References

[1] A. Alfakih. Graph rigidity via Euclidean distance matrices. *Linear Algebra and its Applications*, 310:149–165, 2000.

[2] A. Alfakih. On rigidity and realizability of weighted graphs. *Linear Algebra and its Applications*, 325:57–70, 2001.

[3] A. Alfakih and H. Wolkowicz. On the embeddability of weighted graphs in Euclidean spaces. Technical Report, CORR 98-12, Department of Combinatorics and Optimization, University of Waterloo, 1998. Available at http://orion.math.uwaterloo.ca/˜hwolkowi/.

[4] A. Alfakih, A. Khandani, and H. Wolkowicz. Solving Euclidean distance matrix completion problems via semidefinite programming. *Computational Optimization and Applications*, 12:13–30, 1999.

[5] F. Alizadeh. Interior point methods in semidefinite programming with applications in combinatorial optimization. *SIAM Journal on Optimization*, 5:13–51, 1995.

[6] N. Alon and N. Kahale. Approximating the independence number via the ϑ-function. *Mathematical Programming*, 80:253–264, 1998.

[7] N. Alon and B. Sudakov. Bipartite subgraphs and the smallest eigenvalue. *Combinatorics, Probability and Computing*, 9:1–12, 2000.

[8] N. Alon, B. Sudakov and U. Zwick. Constructing worst case instances for semidefinite programming based approximation algorithms. *SIAM Journal on Discrete Mathematics*, 15:58–72, 2002.

[9] M.F. Anjos. *New convex relaxations for the maximum cut and VLSI layout problems.* PhD thesis, University of Waterloo, 2001.

[10] M.F. Anjos and H. Wolkowicz. A strengthened SDP relaxation via a second lifting for the Max-Cut problem. *Discrete Applied Mathematics*, 119:79–106, 2002.

[11] M.F. Anjos and H. Wolkowicz. Geometry of semidefinite Max-Cut relaxations via ranks. Report CORR 2001-39, Waterloo University, Canada, 2001. [*Journal of Combinatorial Optimization*, to appear.] Available at http://orion.math.uwaterloo.ca:80/˜hwolkowi/.

[12] K. Anstreicher, and N. Brixius. A lower bound for the Quadratic Assignment Problem based on Convex Quadratic Programming. *Mathematical Programming*, 89:341–357, 2001.

[13] K. Anstreicher, N. Brixius, J.-P. Goux and J. Linderoth. Solving large quadratic assignment problems on computational grids. *Mathematical Programming B*, 91:563–588, 2002.

[14] K. Anstreicher and H. Wolkowicz. On Lagrangian relaxation of quadratic matrix constraints. *SIAM Journal on Matrix Analysis and its Applications*, 22:41–55, 2000.

[15] S. Arora, D. Karger and M. Karpinski. Polynomial time approximation schemes for dense instances of NP-hard problems. In *Proceedings of the 27th Annual ACM Symposium on Theory of Computing*, ACM, New York, pages 284–293, 1995.

[16] S. Arora, C. Lund, R. Motwani, M. Sudan, and M. Szegedy. Proof verification and intractability of approximation problems. In *Proceedings of the 33rd IEEE Symposium on Foundations of Computer Science*, IEEE Computer Science Press, Los Alamitos, CA, pp. 14–23, 1992.

[17] E. Balas. Disjunctive programming. *Annals of Discrete Mathematics*, 5:3–51, 1979.

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

[19] M.O. Ball, W. Liu, and W.R. Pulleyblank. Two terminal Steiner tree polyhedra. In *Contributions to Operations research and Economics*, B. Tulkens and H. Tulkens, eds., MIT Press, Cambridge, MA, pp. 251–284, 1989.

[20] F. Barahona. On cuts and matchings in planar graphs. *Mathematical Programming*, 60:53–68, 1993.

[21] F. Barahona. On the computational complexity of Ising spin glass models. *Journal of Physics A, Mathematical and General*, 15:3241–3253, 1982.

[22] F. Barahona. The max-cut problem on graphs not contractible to K_5. *Operations Research Letters*, 2:107–111, 1983.

[23] F. Barahona and A.R. Mahjoub. On the cut polytope. *Mathematical Programming*, 36:157–173, 1986.

[24] F. Barahona and A.R. Mahjoub. Compositions of graphs and polyhedra. II: Stable sets. *SIAM Journal on Discrete Mathematics*, 7:359-371, 1994.

[25] A.I. Barvinok. Feasibility testing for systems of real quadratic equations. *Discrete and Computational Geometry,* 10:1–13, 1993.

[26] A.I. Barvinok. Problems of distance geometry and convex properties of quadratic maps. *Discrete and Computational Geometry*, 13:189–202, 1995.

[27] A.I. Barvinok. A remark on the rank of positive semidefinite matrices subject to affine constraints. *Discrete and Computational Geometry*, 25:23–31, 2001.

[28] S. Basu, R. Pollack and M.-F. Roy. On the combinatorial and algebraic complexity of quantifier elimination. *Journal of the Association for Computing Machinery*, 43:1002–1045, 1996.

[29] M. Bellare and P. Rogaway. The complexity of approximating a nonlinear program. *Mathematical Programming*, 69:429–441, 1995.

[30] C. Berge. Sur une conjecture relative au problème des codes optimaux. Communication, 13ème assemblée générale de l'URSI, Tokyo, 1962.

[31] P. Berman and M. Karpinski. On some tighter inapproximability results, further improvements. *Electronic Colloquium on Computational Complexity*, Report TR98-065, 1998. Available at http://www.eccc.uni-trier.de/eccc/.

[32] A. Blum. New approximation algorithms for graph colouring. *Journal of the Association for Computing Machinery*, 41:470–516, 1994.

[33] J. Bochnak, M. Coste and M.-F. Roy. *Géométrie Algébrique Réelle.* Springer-Verlag, 1987.

[34] A. Bockmayr, F. Eisenbrand, M. Hartmann, and A.S. Schulz. On the Chvátal rank of polytopes in the 0/1 cube. *Discrete Applied Mathematics*, 98:21–27, 1999.

[35] I.M. Bomze, M. Dür, E. de Klerk, C. Roos, A.J. Quist, and T. Terlaky. On copositive programming and standard quadratic optimization problems. *Journal of Global Optimization*, 18:301–320, 2000.

[36] I.M. Bomze and E. de Klerk. Solving standard quadratic optimization problems via semidefinite and copositive programming. Technical report TR 2001-03, ISDS, Universität Wien. [*Journal of Global Optimization*, to appear.]

[37] J.M. Borwein and H. Wolkowicz. Regularizing the abstract convex program. *Journal of Mathematical Analysis and Applications*, 83:495–530, 1981.

[38] J. Bourgain. On Lipschitz embedding of finite metric spaces in Hilbert space. *Israel Journal of Mathematics*, 52:46–52, 1985.

[39] A. Caprara and A.N. Letchford. On the separation of split cuts and related inequalities. Preprint, 2000. [*Mathematical Programming*, to appear.]
Available at http://www.lancs.ac.uk/staff/letchfoa/home.htm.

[40] E. Cela. The Quadratic Assignment Problem: Theory and Algorithms. Kluwer Academic Publishers, USA, 1998.

[41] S. Ceria. *Lift-and-Project Methods for Mixed 0-1 Programs.* Ph.D. dissertation, Graduate School of Industrial Administration, Carnegie Mellon University, US, 1993.

[42] S. Ceria and G. Pataki. Solving integer and disjunctive programs by lift-and-project. In R.E. Bixby, E.A. Boyd, and R.Z. Rios-Mercato, eds., IPCO VI, *Lecture Notes in Computer Science*, 1412:271–283, 1998.

[43] V. Chvátal. Edmonds polytopes and a hierarchy of combinatorial problems. *Discrete Mathematics*, 4:305–337, 1973.

[44] V. Chvátal. On certain polytopes associated with graphs. *Journal of Combinatorial Theory B*, 18:138–154, 1975.

[45] V. Chvátal, W. Cook, and M. Hartman. On cutting-plane proofs in combinatorial optimization. *Linear Algebra and its Applications*, 114/115:455–499, 1989.

[46] W. Cook and S. Dash. On the matrix-cut rank of polyhedra. *Mathematics of Operations Research*, 26:19–30, 2001.

[47] W. Cook, R. Kannan, and A. Schrijver. Chvátal closures for mixed integer programming problems. *Mathematical Programming*, 47:155–174, 1990.

[48] G. Cornuéjols and Y. Li. Elementary closures for integer programs. *Operations Research Letters,* 28:1–8, 2001.

[49] G. Cornuéjols and Y. Li. On the rank of mixed 0-1 polyhedra. In K. Aardal and A.M.H. Gerards, eds., IPCO 2001, *Lecture Notes in Computer Science*, 2081:71–77, 2001.

[50] G.M. Crippen and T.F. Havel. *Distance Geometry and Molecular Conformation.* Research Studies Press, Taunton, Somerset, England, 1988.

[51] D. Cvetković, M. Cangalvić, and V. Kovačevič-Vujčić, *Semidefinite relaxations of traveling salesman problem*, Tech. report, University of Belgrade, Belgrade, Yugoslavia, 1999, to be presented at IPCO 1999 in Graz.

[52] S. Dash. *On the matrix cuts of Lovász and Schrijver and their use in integer programming.* PhD thesis, Rice University, 2001.

[53] C. Delorme and S. Poljak. Laplacian eigenvalues and the maximum cut problem. *Mathematical Programming*, 62:557–574, 1993.

[54] C. Delorme and S. Poljak. Combinatorial properties and the complexity of a max-cut approximation. *European Journal of Combinatorics*, 14:313–333, 1993.

[55] C. Delorme and S. Poljak. The performance of an eigenvalue bound on the max-cut problem in some classes of graphs. *Discrete Mathematics*, 111:145–156, 1993.

[56] P. Delsarte. An algebraic approach to the association schemes of coding theory. *Philips Research Reports Supplements*, No. 10, 1973.

[57] M. Deza and M. Laurent. *Geometry of Cuts and Metrics.* Springer-Verlag, 1997.

[58] R.J. Duffin. Infinite Programs, in: H.W. Kuhn and A.W. Tucker eds., *Linear Inequalities and Related Systems*, Annals of Mathematical Studies vol. 38, Princeton University Press, 1956, 157–170.

[59] F. Eisenbrand. On the membership problem for the elementary closure of a polyhedron. *Combinatorica*, 19:299-300, 1999.

[60] F. Eisenbrand and A.S. Schulz. Bounds on the Chvátal rank of polytopes in the 0/1 cube. In G. Cornuéjols et al., eds, IPCO 1999, *Lecture Notes in Computer Science*, 1610:137–150, 1999.

[61] U. Feige. Randomized graph products, chromatic numbers, and the Lovász ϑ-function. *Combinatorica*, 17: 79–90, 1997.

[62] U. Feige. Randomized rounding of semidefinite programs – variations on the MAX CUT example. Randomization, Approximation, and Combinatorial Optimization, Proceedings of Random-Approx'99, *Lecture Notes in Computer Science*, 1671:189–196, Springer-Verlag, 1999.

[63] U. Feige and M. Goemans. Approximating the value of two prover proof systems, with applications to MAX 2SAT and MAX DICUT. *Proceedings of the 3rd Israel Symposium on the Theory of Computing and Systems*, 182–189, 1995.

[64] U. Feige, M. Karpinski, and M. Langberg. Improved approximation of max-cut on graphs of bounded degree. *Electronic Colloquium on Computational Complexity*, Report TR00-021, 2000. Available at http://www.eccc.uni-trier.de/eccc/.

[65] U. Feige, M. Karpinski, and M. Langberg. A note on approximating max-bisection on regular graphs. *Electronic Colloquium on Computational Complexity*, Report TR00-043, 2000. Available at http://www.eccc.uni-trier.de/eccc/.

[66] U. Feige and G. Schechtman. On the optimality of the random hyperplane rounding technique for MAX CUT. Preprint, 2000. [*Random Structures and Algorithms*, to appear.] Available at http://www.wisdom.weizmann.ac.il/~feige/.

[67] M. Fiedler, Bounds for eigenvalues of doubly stochastic matrices, *Linear Algebra and its Applications*, 5:299–310, 1972.

[68] M. Fiedler, Algebraic connectivity of graphs. *Czechoslovak Mathematical Journal*, 23:298–305, 1973.

[69] P. Frankl and V. Rödl. Forbidden intersections. *Transactions of the American Mathematical Society*, 300:259–286, 1987.

[70] A. Frieze and M. Jerrum. Improved approximation algorithms for MAX k-CUT and MAX BISECTION. *Algorithmica*, 18:67–81, 1997.

[71] T. Fujie and M. Kojima. Semidefinite programming relaxation for nonconvex quadratic programs. *Journal of Global Optimization*, 10:367–380, 1997.

[72] D.R. Fulkerson. Anti-blocking polyhedra. *Journal of Combinatorial Theory B*, 12:50–71, 1972.

[73] M.R. Garey, D.S. Johnson and L. Stockmeyer. Some simplified NP-complete graph problems. *Theoretical Computer Science*, 1:237–267, 1976.

[74] M. Goemans and F. Rendl. Semidefinite programs and association schemes. *Computing*, 63:331–340, 1999.

[75] M.X. Goemans and L. Tunçel. When does the positive semidefiniteness constraint help in lifting procedures? *Mathematics of Operations Research*, 26:796–815, 2001.

[76] M.X. Goemans and D.P. Williamson. New 3/4-approximation algorithms for the maximum satisfiability problem. *SIAM Journal on Discrete Mathematics*, 7:656–666, 1994.

[77] M.X. Goemans and D.P. Williamson. Improved approximation algorithms for maximum cuts and satisfiability problems using semidefinite programming. *Journal of the Association for Computing Machinery*, 42:1115–1145, 1995.

[78] M.X. Goemans and D.P. Williamson. Approximation algorithms for MAX-3-CUT and other problems via complex semidefinite programming. In *Proceedings of the 33rd Annual ACM Symposium on Theory of Computing*, ACM, New York, pp. 443–452, 2001.

[79] 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.

[80] M. Grötschel, L. Lovász, and A. Schrijver. *Geometric Algorithms and Combinatorial Optimization.* Springer-Verlag, Berlin, New York, 1988.

[81] M. Grötschel and W.R. Pulleyblank. Weakly bipartite graphs and the max-cut problem. *Operations Research Letters*, 1:23–27, 1981.

[82] G. Gruber and F. Rendl. Computational experience with stable set relaxations. Preprint, 2001. Available at http://www-sci.uni-klu.ac.at/math-or/home/publicat.htm.

[83] B. Guenin. A characterization of weakly bipartite graphs. *Journal of Combinatorial Theory B*, 83:112–168, 2001.

[84] S.W. Hadley, F. Rendl, and H. Wolkowicz. A new lower bound via projection for the quadratic assignment problem. *Mathematics of Operations Research*, 17:727–739, 1992.

[85] E. Halperin, D. Livnat, and U. Zwick. MAX-CUT in cubic graphs. In *Proceedings of 13th SODA*, 506–513, 2002.

[86] E. Halperin and U. Zwick. A unified framework for obtaining improved approximations algorithms for maximum graph bisection problems. In K. Aardal and A.M.H. Gerards, eds., IPCO 2001, *Lecture Notes in Computer Science*, 2081:210–225, 2001.

[87] E. Halperin and U. Zwick. Approximation algorithms for MAX 4-SAT and rounding procedures for semidefinite programs. *Journal of Algorithms*, 40:184–211, 2001.

[88] E. Halperin and U. Zwick. Combinatorial approximation algorithms for the maximum directed cut problem. In *Proceedings of 12th SODA*, pages 1–7, 2001.

[89] J. Håstad. Some optimal inapproximability results. In *Proceedings of the 29th Annual ACM Symposium on the Theory of Computing*, ACM, New York, pages 1–10, 1997.

[90] C. Helmberg, F. Rendl, R.J. Vanderbei and H. Wolkowicz. An interior-point method for semidefinite programming. *SIAM Journal on Optimization*, 6:342–361, 1996.

[91] C. Helmberg, F. Rendl and R. Weismantel. A semidefinite programming approach to the quadratic knapsack problem. *Journal of Combinatorial Optimization*, 4:197–215, 2000.

[92] R.D. Hill and S.R. Waters. On the cone of positive semidefinite matrices. *Linear Algebra and its Applications*, 90:81–88, 1987.

[93] A.J. Hoffman and H.W. Wielandt. The variation of the spectrum of a normal matrix. *Duke Mathematical Journal*, 20:37–39, 1953.

[94] R.A. Horn, and C.R. Johnson. *Matrix Analysis.* Cambridge University Press, 1985.

[95] K. Jansen, M. Karpinski, and A. Lingas. A polynomial time approximation scheme for MAX-BISECTION on planar graphs. *Electronic Colloquium on Computational Complexity*, Report TR00-064, 2000. Available at http://www.eccc.uni-trier.de/eccc/.

[96] C.R. Johnson. Matrix completion problems: a survey. In C.R. Johnson, editor, *Matrix Theory and Applications*, volume 40 of *Proceedings of Symposia in Applied Mathematics*, pages 171–198. American Mathematical Society, Providence, Rhode Island, 1990.

[97] C.R. Johnson, B. Kroschel, and H. Wolkowicz. An interior-point method for approximate positive semidefinite completions. *Computational Optimization and Applications*, 9:175–190, 1998.

[98] D. Johnson. Approximation algorithms for combinatorial problems. *Journal of Computer and System Sciences*, 9:256–278, 1974.

[99] V. Kann, S. Khanna, J. Lagergren, and A. Panconesi. On the hardness of approximating MAX k-CUT and its dual. *Chicago Journal of Theoretical Computer Science*, 2, 1997.

[100] D. Karger, R. Motwani, and M. Sudan. Approximate graph colouring by semidefinite programming. *Journal of the Association for Computing Machinery*, 45:246–265, 1998.

[101] H. Karloff. How good is the Goemans-Williamson max-cut algorithm? *SIAM Journal on Computing*, 29:336–350, 1999.

[102] H. Karloff and U. Zwick. A 7/8-approximation algorithm for MAX 3SAT? In *Proceedings of the 38th Annual IEEE Symposium on Foundations of Computer Science*, Miami Beach, Florida, pp. 406–415, 1997.

[103] R.M. Karp. Reducibility among combinatorial problems. In *Complexity of Computer Computations*, Plenum Press, New York, pp. 85–103, 1972.

[104] L. Khachiyan and L. Porkolab. Computing integral points in convex semi-algebraic sets. In *38th Annual Symposium on Foundations of Computer Science*, pages 162–171, 1997.

[105] L. Khachiyan and L. Porkolab. Integer optimization on convex semialgebraic sets. *Discrete and Computational Geometry*, 23:207–224, 2000.

[106] S. Khanna, N. Linial and S. Safra. On the hardness of approximating the chromatic number. *Combinatorica*, 20:393–415, 2000.

[107] J. Kleinberg and M.X. Goemans. The Lovász theta function and a semidefinite programming relaxation of vertex cover. *SIAM Journal on Discrete Mathematics*, 11:196–204, 1998.

[108] E. de Klerk, D.V. Pasechnik, and J.P. Warners. Approximate graph colouring and MAX-k-CUT algorithms based on the ϑ-function. Preprint, 2000. Available at http://ssor.twi.tudelft.nl/˜ deklerk/.

[109] E. de Klerk and D.V. Pasechnik. On the performance guarantee of MAX 3-CUT approximation algorithms. Preprint, 2001.

[110] E. de Klerk and D.V. Pasechnik. Approximation of the stability number of a graph via copositive programming. *SIAM Journal on Optimization*, 12:875–892, 2002.

[111] D.E. Knuth. The sandwich theorem. *Electronic Journal of Combinatorics*, 1:1–48, 1994.

[112] M. Kojima, S. Shindoh and S. Hara. Interior-point methods for the monotone semidefinite linear complementarity problem in symmetric matrices. *SIAM Journal on Optimization*, 7:86–125, 1997.

[113] M. Kojima and L. Tunçel. Cones of matrices and successive convex relaxations of nonconvex sets. *SIAM Journal on Optimization*, 10:750–778, 2000.

[114] J.B. Lasserre. Optimality conditions and LMI relaxations for $0-1$ programs. Technical Report N. 00099, LAAS, Toulouse, 2000.

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

[116] J.B. Lasserre. An explicit exact SDP relaxation for nonlinear $0-1$ programs. In K. Aardal and A.M.H. Gerards, eds., IPCO 2001, *Lecture Notes in Computer Science*, 2081:293–303, 2001.

[117] J.B. Lasserre. SDP versus LP-relaxations for polynomial programming. Preprint, 2001. [*Mathematics of Operations Research*, to appear.]

[118] M. Laurent. The real positive semidefinite completion problem for series-parallel graphs. *Linear Algebra and its Applications*, 252:347–366, 1997.

[119] M. Laurent. A connection between positive semidefinite and Euclidean distance matrix completion problems. *Linear Algebra and its Applications*, 273:9–22, 1998.

[120] M. Laurent. A tour d'horizon on positive semidefinite and Euclidean distance matrix completion problems. In P. Pardalos and H. Wolkowicz, editors, *Topics in Semidefinite and Interior-Point Methods*, vol. 18 of the Fields Institute for Research in Mathematical Science, Communication Series, Providence, Rhode Island, pages 51–76, 1998.

[121] M. Laurent. Polynomial instances of the positive semidefinite and Euclidean distance matrix completion problems. *SIAM Journal on Matrix Analysis and its Applications,* 22:874–894, 2000.

[122] M. Laurent. On the sparsity order of a graph and its deficiency in chordality. *Combinatorica*, 21:543–570, 2001.

[123] M. Laurent. Tighter linear and semidefinite relaxations for max-cut based on the Lovász-Schrijver lift-and-project procedure. *SIAM Journal on Optimization*, 12:345–375, 2001.

[124] M. Laurent. A comparison of the Sherali-Adams, Lovász-Schrijver and Lasserre relaxations for $0,1$-programming. Report PNA-R0108, CWI, Amsterdam, 2001. Available at http://www.cwi.nl/˜ monique/.

[125] M. Laurent. Semidefinite relaxations for Max-Cut. To appear in *The Sharpest Cut, Festschrift in Honour of M. Padberg's 60th Birthday*, M. Grötschel, ed., SIAM. Available at http://www.cwi.nl/˜ monique/.

[126] M. Laurent and S. Poljak. On a positive semidefinite relaxation of the cut polytope. *Linear Algebra and its Applications*, 223/224:439–461, 1995.

[127] M. Laurent and S. Poljak. On the facial structure of the set of correlation matrices. *SIAM Journal on Matrix Analysis and its Applications*, 17:530–547, 1996.

[128] H.W. Lenstra, Jr.. Integer programming with a fixed number of variables. *Mathematics of Operations Research*, 8:538–548, 1983.

[129] N. Linial, E. London, and Yu. Rabinovich. The geometry of graphs and some of its algorithmic consequences. *Combinatorica*, 15:215–245, 1995.

[130] N. Linial, A. Magen, and A. Naor. Girth and Euclidean distortion. Preprint, 2001. Available at http://www.cs.huji.ac.il/˜nati/.

[131] N. Linial and M.E. Sachs. On the Euclidean distortion of complete binary trees. Preprint, 2001. [*Discrete and Computational Geometry*, to appear.] Available at http://www.cs.huji.ac.il/˜nati/.

[132] W. Liu. *Extended Formulations and Polyhedral Projection.* Ph.D. thesis, Department of Combinatorics and Optimization, University of Waterloo, Canada, 1988.

[133] L. Lovász. Normal hypergraphs and the perfect graph conjecture. *Discrete Mathematics*, 2:253–267, 1972.

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

[135] L. Lovász. Stable sets and polynomials. *Discrete Mathematics*, 124:137–153, 1994.

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

[137] C. Lund and M. Yannakakis. On the hardness of approximating minimization problems. In *Proceedings of the 25th Annual ACM Symposium on Theory of Computing*, ACM, New York, pp. 286–293, 1993.

[138] N. Maculan. The Steiner problem in graphs. *Annals of Discrete Mathematics*, 31:185–222, 1987.

[139] S. Mahajan and H. Ramesh. Derandomizing semidefinite programming based approximation algorithms. In *Proceedings of the 36th Symposium on Foundations of Computer Science*, IEEE Computer Science Press, Los Alamitos, CA, pp. 162–169, 1995.

[140] S. Matuura and T. Matsui. 0.863-approximation algorithm for MAX DICUT. In M. Goemans et als., eds., APPROX 2001 and RANDOM 2001, *Lecture Notes in Computer Science,* 2129:138–146, 2001.

[141] S. Matuura and T. Matsui. 0.935-approximation randomized algorithm for MAX 2SAT and its derandomization. Technical Report METR 2001-03, University of Tokyo, 2001. Available at http://www.keisu.t.u-tokyo.ac.jp/METR.html.

[142] R.J. McEliece, E.R. Rodemich and H.C. Rumsey, Jr. The Lovász bound and some generalizations. *Journal of Combinatorics and System Sciences*, 3:134–152, 1978.

[143] P. Meurdesoif. Strenghtening the Lovász $\vartheta(\overline{G})$ bound for graph colouring. Preprint, 2000. [*Mathematical Programming*, to appear.]

[144] B. Mohar and S. Poljak. Eigenvalues and the max-cut problem. *Czechoslovak Mathematical Journal*, 40:343–352, 1990.

[145] R.D.C. Monteiro. Primal-dual path-following algorithms for semidefinite programming. *SIAM Journal on Optimization*, 7:663–678, 1997.

[146] T.S. Motzkin and E.G. Straus. Maxima for graphs and a new proof of a theorem of Túran. *Canadian Journal of Mathematics*, 17:533–540, 1965.

[147] K.G. Murty and S.N. Kabadi. Some NP-complete problems in quadratic and linear programming. *Mathematical Programming*, 39:117–129, 1987.

[148] G. Nemhauser and L. Wolsey. *Integer and Combinatorial Optimization.* John Wiley and Sons, New York, 1988.

[149] Y. Nesterov. Quality of semidefinite relaxation for nonconvex quadratic optimization. CORE Discussion Paper # 9719, Belgium, 1997.

[150] Y. Nesterov. Semidefinite relaxation and nonconvex quadratic optimization. *Optimization Methods and Software*, 9:141–160, 1998.

[151] Y. Nesterov. Squared functional systems and optimization problems. In J.B.G. Frenk, C. Roos, T. Terlaky, and S. Zhang, eds., *High Performance Optimization*, pages 405–440. Kluwer Academic Publishers, 2000.

[152] J. von Neumann. Some matrix inequalities and metrization of matrix space. *Tomsk Univ. Rev.* 1:286–300, 1937. (reprinted in: John von Neumann: Collected works, vol. 4, A.H. Taub ed., MacMillan, 205–219, 1962.)

[153] C.E. Nugent, T.E. Vollman, and J. Ruml. An experimental comparison of techniques for the assignment of facilities to locations. *Operations Research*, 16:150–173, 1968.

[154] M.L. Overton and R.S. Womersley. On the sum of the largest eigenvalues of a symmetric matrix. *SIAM Journal on Matrix Analysis and its Applications*, 13:41–45, 1992.

[155] P.A. Parrilo. *Structured semidefinite programs and semialgebraic geometry methods in robustness and optimization.* PhD thesis, California Institute of Technology, May 2000.

[156] P.A. Parrilo. Semidefinite programming relaxations for semialgebraic problems. Preprint, 2001. Available at http://www.cds.caltech.edu/˜pablo.

[157] P.A. Parrilo and B. Sturmfels. Minimizing polynomial functions. Preprint, 2001. Available at http://www.arxiv.org/abs/math.OC/0103170.

[158] G. Pataki. Cone-LP's and semidefinite programs: Geometry and a simplex-type method. In W.H. Cunningham, S.T. MacCormick and M. Queyranne, eds., IPCO 1996, *Lecture Notes in Computer Science*, 1084:162–174, 1996.

[159] G. Pataki. On the rank of extreme matrices in semidefinite programs and the multiplicity of optimal eigenvalues. *Mathematics of Operations Research*, 23:339–358, 1998.

[160] S. Poljak. Polyhedral and eigenvalue approximations of the max-cut problem. In *Sets, Graphs, and Numbers*, vol. 60 of *Colloquia Mathematica Societatis János Bolyai*, pages 569–581. Budapest, Hungary, 1991.

[161] S. Poljak and F. Rendl. Nonpolyhedral relaxation sof graph-bisection problems. *SIAM Journal on Optimization*, 5:467–487, 1995.

[162] S. Poljak and Z. Tuza. The expected relative error of the polyhedral approximation of the max-cut problem. *Operations Research Letters*, 16:191–198, 1994.

[163] S. Poljak and Z. Tuza. Maximum cuts and largest bipartite subgraphs. In W. Cook, L. Lovász and P. Seymour, editors, *Combinatorial Optimization*, vol. 20 of *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, pages 181–244. American Mathematical Society, Providence, RI, 1995.

[164] L. Porkolab and L. Khachiyan. On the complexity of semidefinite programs. *Journal of Global Optimization*, 10:351–365, 1997.

[165] V. Powers and B. Reznick. A new bound for Pólya's Theorem with applications to polynomials positive on polyhedra. Preprint, 2000. [*Journal of Pure and Applied Algebra,* to appear.] Available at http://www.mathcs.emory.edu/˜vicki/preprint/.

[166] V. Powers and T. Wörmann. An algorithm for sums of squares of real polynomials. *Journal of Pure and Applied Algebra*, 127:99–104, 1998.

[167] M. Putinar. Positive polynomials on compact semi-algebraic sets. *Indiana University Mathematics Journal*, 42:969–984, 1993.

[168] A.J. Quist, E. de Klerk, C. Roos and T. Terlaky. Copositive relaxation for general quadratic programming. *Optimization Methods and Software*, 9:185–209, 1998.

[169] M.V. Ramana. An exact duality theory for semidefinite programming and its complexity implications. *Mathematical Programming*, 77:129–162, 1997.

[170] M.V. Ramana and A. Goldman. Some geometric results in semidefinite programming. *Journal of Global Optimization*, 7:33–50, 1995.

[171] M.V. Ramana, L. Tunçel, and H. Wolkowicz. Strong duality for semidefinite programming. *SIAM Journal on Optimization*, 7:641–662, 1997.

[172] B.A. Reed and A.J.L. Ramirez. *Perfect Graphs.* Wiley, 2001.

[173] J. Renegar. On the computational complexity and geometry of the first order theory of the reals. *Journal of Symbolic Computation*, 13(3):255–352, 1992.

[174] A. Schrijver. A comparison of the Delsarte and Lovász bounds. *IEEE Transactions on Information Theory*, IT-25:425–429, 1979.

[175] A. Schrijver. *Theory of Linear and Integer Programming.* John Wiley and Sons, New York, 1986.

[176] A. Schrijver. *Combinatorial Optimization - Polyhedra and Efficiency.* Springer-Verlag, to appear.

[177] P.D. Seymour. The matroids with the max-flow min-cut property. *Journal of Combinatorial Theory B*, 23:189–222, 1977.

[178] H. Sherali and W. Adams. A hierarchy of relaxations between the continuous and convex hull representations for zero-one programming problems. *SIAM Journal on Discrete Mathematics*, 3:411–430, 1990.

[179] H. Sherali and W. Adams. A hierarchy of relaxations and convex hull representations for mixed-integer zero-one programming problems. *Discrete Applied Mathematics*, 52:83–106, 1994.

[180] H. Sherali and W. Adams. *A Reformulation-Linearization Technique (RLT) for Solving Discrete and Continuous Nonconvex Problems.* Kluwer, 1997.

[181] H. Sherali and C.H. Tuncbilek. A global optimization algorithm for polynomial programming problems using a Reformulation-Linearization Technique. *Journal of Global Optimization*, 2:101–112, 1992.

[182] H.D. Sherali and C.H. Tuncbilek. Reformulation-Linearization/Convexification relaxations for univariate and multivariate polynomial programming problems. *Operations Research Letters*, 21:1–10, 1997.

[183] N.Z. Shor. An approach to obtaining global extremums in polynomial mathematical programming problems. *Kibernetika*, 5:102–106, 1987.

[184] N.Z. Shor. Class of global minimum bounds of polynomial functions. *Cybernetics*, 6:731–734, 1987. [Translated from *Kibernetika*, 6:9–11, 1987.]

[185] N.Z. Shor. *Nondifferentiable Optimization and Polynomial Problems.* Kluwer Academic Publishers, 1998.

[186] R. Sotirov. Bundle methods in combinatorial optimization. Dissertation, University of Klagenfurt, forthcoming.

[187] G. Stengle. A Nullstellensatz and a Positivstellensatz in semialgebraic geometry. *Mathematische Annalen*, 207:87–97, 1974.

[188] T. Stephen and L. Tunçel. On a representation of the matching polytope via semidefinite liftings. *Mathematics of Operations Research*, 24:1–7, 1999.

[189] M.J. Todd. A study of search directions directions in primal-dual interior-point methods for semidefinite programming. *Optimization Methods and Software*, 11:1–46, 1999.

[190] L. Trevisan, G.B. Sorkin, M. Sudan, and D.P. Williamson. Gadgets, approximation, and linear programming. In *Proceedings of the 37th Annual IEEE Symposium on Foundations of Computer Science*, Burlington, Vermont, pp. 617–626, 1996.

[191] P. Tseng. Further results on approximating nonconvex quadratic optimization by semidefinite programming relaxation. Preprint, 2001. Available at http://www.optimization-online.org/DB_HTML/2001/12/417.html.

[192] W.F. de la Vega. MAX-CUT has a randomized approximation scheme in dense graphs. *Random Structures and Algorithms*, 8:187–198, 1996.

[193] A. Wigderson. Improving the performance guarantee for approximate graph colouring. *Journal of the Association for Computing Machinery*, 30:729–735, 1983.

[194] H. Wolkowicz, R. Saigal, and L. Vandeberghe (eds.). *Handbook of Semidefinite Programming.* Kluwer, 2000.

[195] M. Yannakakis. Expressing combinatorial optimization problems by linear programs. In *Proceedings of the 29th International IEEE Symposium on Foundations of Computer Science*, White Plains, NY, pp. 223–228, 1988.

[196] M. Yannakakis. On the approximation of maximum satisfiability. *Journal of Algorithms*, 17:475–502, 1994.

[197] Y. Ye. Approximating quadratic programming with bound and quadratic constraints. *Mathematical Programming*, 84:219–226, 1999.

[198] Y. Ye. A 0.699-approximation algorithm for Max-Bisection. *Mathematical Programming*, 90:101–111, 2001.

[199] Y. Zhang. On extending some primal-dual interior-point algorithms from linear programming to semidefinite programming. *SIAM Journal on Optimization*, 8:365–386, 1998.

[200] S. Zhang. Quadratic minimization and semidefinite relaxation. *Mathematical Programming*, 87:453–465, 2000.

[201] Q. Zhao, S.E. Karisch, F. Rendl, and H. Wolkowicz. Semidefinite programming relaxations for the Quadratic Assignment Problem. *Journal of Combinatorial Optimization*, 2:71-109, 1998.

[202] U. Zwick. Outward rotations: a tool for rounding solutions of semidefinite programming relaxations, with applications to MAX CUT and other problems. In *Proceedings of the 31st Annual ACM Symposium on Theory of Computing*, ACM, New York, pp. 679–687, 1999.

[203] U. Zwick. Analyzing the MAX 2-SAT and MAX DI-CUT approximation algorithms of Feige and Goemans. Preprint, 2000. Available at http://www.math.tau.ac.il/~zwick/.

[204] U. Zwick. Computer assisted proof of optimal approximability results. In *Proceedings of 13th SODA*, 496–505, 2002. Available at http://www.math.tau.ac.il/~zwick/.