# MATRIX RELAXATIONS IN COMBINATORIAL OPTIMIZATION

FRANZ RENDL, May 5, 2010 *

**Abstract.** The success of interior-point methods to solve semidefinite optimization problems (SDP) has spurred interest in SDP as a modeling tool in various mathematical fields, in particular in combinatorial optimization. SDP can be viewed as a matrix relaxation of the underlying 0-1 optimization problem. In this survey, some of the main techniques to get matrix relaxations of combinatorial optimization problems are presented. These are based either on semidefinite matrices, leading in general to tractable relaxations, or on completely positive or copositive matrices. Copositive programs are intractable, but they can be used to get exact formulations of many NP-hard combinatorial optimization problems. It is the purpose of this survey to show the potential of matrix relaxations.

**Key words.** Semidefinite Optimization, Lift-and-Project, integer programming.

**1. Introduction.** Integer programming and nonlinear optimization developed and grew rather independently of one another for a long time. The theoretical basis of integer programming consisted essentially of polyhedral combinatorics and the algorithmic machinery for linear programming, while nonlinear optimization relies on local analysis based on vector calculus (Taylor expansion, steepest descent principle, etc). In the last 15 years these two fields mutually opened up, and today interior-point methods are a widely accepted tool in integer programming, while the modeling power of 0-1 decision variables in an otherwise continuous setting expands the flexibility of real-world modeling substantially.

In this article we explore the idea of matrix liftings joining integer and nonlinear optimization. We first introduce the 0-1 formulation of an abstract combinatorial optimization problem (COP), given as follows. Let $E$ be a finite set and let $\mathcal{F}$ be a (finite) family of subsets of $E$. The elements $F \in \mathcal{F}$ represent the feasible solutions of (COP). Each $e \in E$ has a given integer cost $c_e$. We define the cost $c(F)$ of $F \in \mathcal{F}$ to be $c(F) := \sum_{e \in F} c_e$. The problem (COP) now consists in finding a feasible solution $F$ of minimum cost:

$$(COP) \quad z^* = \min\{c(F) : F \in \mathcal{F}\}.$$

The 0-1 model of (COP) is obtained by assigning to each $F \in \mathcal{F}$ a characteristic vector $x_F \in \{0,1\}^{|E|}$ with $(x_F)_e = 1$ if and only if $e \in F$. We can write (COP) as a linear program as follows. Let

$$\mathcal{P} := conv\{x_F : F \in \mathcal{F}\}$$

* Institut für Mathematik, Alpen-Adria-Universität Klagenfurt, Austria.

denote the convex hull of the characteristic vectors of feasible solutions. Then it is clear that

$$z^* = \min\{c^T x_F : F \in \mathcal{F}\} = \min\{c^T x : x \in P\}.$$

The first minimization is over the finite set $\mathcal{F}$, the second one is a linear program. This is the basic principle underlying the polyhedral approach to solve combinatorial optimization problems. The practical difficulty lies in the fact that in general the polyhedron $\mathcal{P}$ is not easily available. We recall two classical examples to illustrate this point.

As a *nice* example, we consider first the **linear assignment problem**. For a given $n \times n$ matrix $C = (c_{ij})$, it consists of finding a permutation $\phi$ of $N = \{1, \ldots, n\}$, such that $\sum_{i \in N} c_{i\phi(i)}$ is minimized. The set of all such permutations is denoted by $\Pi$. In our general setting, we define the ground set to be $E = N \times N$, the set of all ordered pairs $(i, j)$. Feasible solutions are now given through permutations $\phi$ as $F_\phi \subset E$ such that $F_\phi = \{(i, \phi(i)) : i \in N\}$. In this case the characteristic vector of $F_\phi$ is the permutation matrix $X_\phi$ given by $(X_\phi)_{ij} = 1$ if and only if $j = \phi(i)$. Birkhoff's theorem tells us that the convex hull of the set of permutation matrices $\Pi$ is the set of doubly stochastic matrices $\Omega = \{X : Xe = X^T e = e,\ X \geq 0\}$ [1].

THEOREM 1.1. *conv*$\{X_\phi : \ \phi \in \Pi\} = \Omega$.

Hence we have a simple polyhedral description of $\mathcal{P}$ in this case. Therefore

$$\min\left\{\sum_i c_{i\phi(i)} : \phi \in \Pi\right\} = \min\{\langle C, X\rangle : X \in \Omega\Big\},$$

and the linear assignment problem can be solved as an ordinary linear program.

Unfortunately the set $\mathcal{P}$ does not always have such a nice description. As a second example we consider the **stable set problem**. Given a graph $G = (V, E)$ with vertex set $V = \{1, \ldots, n\}$, the problem is to find $S \subseteq V$, such that no edge joins vertices in $S$ (such sets $S$ are called stable), and $|S|$ is maximized. The ground set here is $V$ and $\mathcal{F}$ consists of all subsets of $V$ which are stable. The characteristic vectors $x \in \mathbb{R}^n$ of the stable sets can be characterized by $x = (x_i)$ with $x_i \in \{0, 1\}$ and

$$x_i + x_j \leq 1 \ \forall ij \in E(G), \tag{1.1}$$

because no stable set can contain both $i$ and $j$ if $ij \in E(G)$. A partial description of the convex hull of the characteristic vectors of stable sets is therefore given by

$$FSTAB(G) := \{x \in \mathbb{R}^n : x \geq 0, x_i + x_j \leq 1 \ \forall ij \in E(G)\}, \tag{1.2}$$

[1] For notation we refer to the end of this section.

leading to the following linear programming relaxation

$$\max\{e^T x : x \in FSTAB(G)\}.$$

If we take $G$ to be the 5-cycle $C_5$, we see that $x = \frac{1}{2}e$ is feasible for $FSTAB(C_5)$ with value $\frac{5}{2}$, showing that this is indeed only a relaxation of the stable set problem.

The use of a computationally tractable partial description of $\mathcal{P}$ by linear inequalities in combination with systematic enumeration, like Branch and Bound, has led to quite successful solution methods for a variety of combinatorial optimization problems like the traveling salesman problem (TSP), see for instance [40]. It turned out however, that for some prominent NP-hard problems like Stable-Set or Max-Cut, this polyhedral approach was not as successful as one might have hoped in view of the results for TSP. It is the purpose of this article to describe matrix based relaxations, which generalize the purely polyhedral methods, and have the potential for stronger approximations of the original problem.

We conclude the introduction with a summary of the **notation** used throughout.

**Vectors and matrices:** $e$ denotes the vector of all ones of appropriate dimension and $J = ee^T$ is the all ones matrix. $I = (e_1, \ldots, e_n)$ denotes the $n \times n$ identity matrix, so the $e_i'$s represent the standard unit vectors. We also use $(\delta_{ij}) := I$, thereby defining the Kronecker delta $\delta_{ij}$. The sum of the main diagonal entries of a square matrix $A$ is called the trace, $tr(A) = \sum_i a_{ii}$. The inner product in $\mathbb{R}^n$ as well as in the space of $n \times n$ matrices is represented by $\langle ., . \rangle$. Hence $\langle a, b \rangle = a^T b$, for $a, b \in \mathbb{R}^n$ and $\langle A, B \rangle = tr(A^T B)$ for matrices $A, B$. The Kronecker product of two matrices $P, Q$ is the matrix consisting of all possible products of elements from $P$ and $Q$, $P \otimes Q = (p_{ij}Q)$. For $m \in \mathbb{R}^n$ we define $Diag(m)$ to be the diagonal matrix having $m$ on the main diagonal. $diag(M)$ is the vector, containing the main diagonal elements from $M$. If $X = (x_1, \ldots, x_n)$ is a matrix with columns $x_i$, then $\mathrm{vec}(X) = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}$ is the vector, obtained by stacking the columns of $X$.

**Sets and matrix cones:** The standard simplex in $\mathbb{R}^n$ is given by $\Delta = \{x \in \mathbb{R}^n : e^T x = 1,\ x \geq 0\}$. In this survey, we will put special emphasis on the following matrix cones.

$$\mathcal{N} := \{X : X \geq 0\} \text{ elementwise nonnegative matrices,}$$

$$\mathcal{S}^+ := \{X : X = X^T, a^T X a \geq 0\ \forall a\}, \text{ positive semidefinite matrices,}$$

$$\mathcal{C} := \{X : X = X^T, a^T X a \geq 0\ \forall a \geq 0\}, \text{ copositive matrices.}$$

We also use the notation $X \succeq 0$ to express $X \in \mathcal{S}^+$. If $K$ is a cone in some finite dimensional vector space $\mathbb{R}^d$, then by definition the dual cone, denoted $K^*$, contains all elements from $\mathbb{R}^d$ having nonnegative inner product with all elements from $K$,

$$K^* := \{y \in \mathbb{R}^d : \langle x, y\rangle \geq 0 \ \forall x \in K\}.$$

It is well known and can easily be shown, that both $\mathcal{N}$ and $\mathcal{S}^+$ are self-dual. The dual cone of $\mathcal{C}$ is the cone

$$C^* = conv\{aa^T : a \geq 0\} = \{Y : \exists X \geq 0,\ Y = XX^T\}$$

of completely positive matrices. Membership in $\mathcal{S}^+$ can be checked in polynomial time, for instance through the existence (or non-existence) of the Cholesky decomposition. In contrast, it is NP-complete to decide if a matrix does not belong to $\mathcal{C}$, see [49]. Wile positive semidefinite matrices are covered in any reasonable text book on advanced linear algebra, we refer the reader to [4] for a thorough treatment of completely positive matrices and to [33] for a recent survey on copositive matrices.

**Graphs:** A graph $G = (V, E)$ is given through the set of vertices $V$ and the set of edges $E$. We sometimes write $E(G)$ to indicate the dependence on $G$. If $S \subset V$, we denote by $\delta(S) := \{uv \in E : u \in S,\ v \notin S\}$ the set of edges joining $S$ and $V \setminus S$. We also say that the edges in $\delta(S)$ are cut by $S$.

**2. Matrix relaxations: basic ideas.** The classical polyhedral approach is formulated as a relaxation in $\mathbb{R}^n$, the natural space to embed $\mathcal{F}$. Here $n$ denotes the cardinality of $E$, $|E| = n$. Matrix-based relaxations of (COP) are easiest explained as follows. To an element $x_F \in \mathcal{F}$ we associate the matrix $x_F x_F^T$ and consider

$$\mathcal{M} := conv\{x_F x_F^T : F \in \mathcal{F}\}, \tag{2.1}$$

see for instance [44, 61, 62]. Note that

$$diag(x_F x_F^T) = x_F,$$

because $x_F \in \{0, 1\}^n$. This property immediately shows that the original linear relaxation, obtained through a partial description of $\mathcal{P}$ can also be modeled in this setting. The full power of matrix lifting is based on the possibility to constrain $\mathcal{M}$ to matrix cones other than polyhedral ones. Moreover, quadratic constraints on $x_F$ will turn into linear constraints on matrices in $\mathcal{M}$.

If $K$ is some matrix cone and matrices $C, A_1, \ldots, A_m$ and $b \in \mathbb{R}^m$ are given, the problem

$$\inf\{\langle C, X\rangle : \langle A_i, X\rangle = b_i \ i = 1, \ldots, m,\ X \in K\} \tag{2.2}$$

is called a linear program over $K$. Linear programs over $\mathcal{S}^+$ are also called semidefinite programs (SDP) and those over $C$ or $C^*$ are called copositive programs (CP) for short. In this paper we will mostly concentrate on SDP and CP relaxations of combinatorial optimization problems.

The duality theory of linear programming generalizes easily to conic linear programs. The (Lagrangian) dual associated to (2.2) is given as

$$\sup\{b^T y : C - \sum_i y_i A_i \in K^*\}. \tag{2.3}$$

Weak duality ($\sup \leq \inf$) holds by construction of the dual. Strong duality ($\sup = \inf$), as well as attainment of the respective optima requires some sort of regularity of the feasible regions. We refer to Duffin [17] for the original paper, and to the handbook [70] on semidefinite programming for a detailed discusssion of SDP. The existence of feasible points in the interior of the primal and dual cone insures the following characterization of optimality. For ease of notation we write $A(X) = b$ for the equations in (2.2). The linear operator $A$ has an adjoint $A^T$, defined through the adjoint identity

$$\langle A(X), y\rangle = \langle X, A^T(y)\rangle.$$

We should point out that the inner product on the left is in $\mathbb{R}^m$ and on the right it is in the space of $n \times n$ matrices. In this paper the inner products will always be canonical, so we do not bother to overload the notation to distinguish them. The adjoint can be expressed as $A^T(y) = \sum_i y_i A_i$.

THEOREM 2.1. *[17, 70] Suppose there exists $X_0 \in int(K)$ such that $A(X_0) = b$ and there is $y_0$ such that $C - A^T(y_0) \in int(K^*)$. Then the optima in (2.2) and (2.3) are attained. Moreover, $X$ and $y$ are optimal if and only if $A(X) = b$, $X \in K$, $Z := C - A^T(y) \in K^*$ and the optimal objective values coincide, $\langle X, Z\rangle = 0$.*

Matrix relaxations can be used in several ways to better understand (COP). The seminal work of Goemans and Williamson [23] opened the way to new approximation techniques for some COPs. We will briefly explain them as we develop the various relaxations. From a computational point of view, SDP based relaxations pose a serious challenge to existing algorithms for SDP. We will describe the currently most efficient ways to solve these relaxations (at least approximately).

There exist several recent survey papers devoted to the connection between semidefinite optimization and integer programming. The interested reader is referred to [39] for an extensive summary on the topic covering the development until 2003. The surveys by Lovász [43], Goemans [22] and Helmberg [29] all focus on the same topic, but also reflect the scientific interests and preferences of the respective authors. The present paper is no exception to this principle. The material selected, and also omitted,

reflects the author's subjective view on the subject. It is a continuation and an extension of [57].

We are now going to look at several techniques to obtain matrix relaxations of combinatorial optimization problems. We start out with the generic idea, as explained in the introduction, and show how it works for graph partitioning.

**3. Graph partition.** Graph partition problems come in various formulations. The starting point is a graph $G$, given through its weighted $n \times n$ adjcacency matrix $A_G$, or simply $A$. If $ij \in E(G)$, then $a_{ij}$ denotes the weight of edge $ij$, otherwise $a_{ij} = 0$. Hence, $A = A^T$ and $diag(A) = 0$. The Laplacian $L_A$ of $A$, or $L$ for short, is defined to be the matrix

$$L := diag(Ae) - A. \tag{3.1}$$

The following simple properties of the Laplacian $L$ will be used later on.

PROPOSITION 3.1. *The Laplacian $L$ of the matrix $A$ satisfies $Le = 0$ and $A \geq 0$ implies that $L \succeq 0$.*

Graph partition problems ask to separate the vertices of a graph into a specified number of partition blocks so that the total weight of edges joining different blocks is minimized or maximized. Partition problems lead rather naturally to matrix based relaxations because encoding whether or not vertices $i$ and $j \in V$ are separated has a natural matrix representation, as we will see briefly. We recall the definition of a cut given by $S \subset V$: $\delta(S) = \{uv \in E(G) : u \in S,\ v \notin S\}$.

**3.1. Max-$k$-Cut.** For $k \geq 2$, Max-k-Cut asks to partition $V(G)$ into $k$ subsets $(S_1, \ldots, S_k)$ such that the total weight of edges joining distinct subsets is maximized. We introduce characteristic vectors $s_i \in \{0,1\}^n$ for each $S_i$. The $n \times k$ matrix $S = (s_1, \ldots, s_k)$ is called the $k$-partition matrix. Since $\cup_i S_i = V$, we have

$$\sum_{i=1}^{k} s_i = Se = e.$$

Partition matrices have the following properties.

PROPOSITION 3.2. *Let $S = (s_1, \ldots, s_k)$ be a $k$-partition matrix. Then $diag(SS^T) = e,\ kSS^T - J \succeq 0$.*

We prove a more general result, which will also be of use later on. Its proof has been pointed out by M. Laurent[2], see also [18], Lemma 2.

LEMMA 3.1. *Let $s_1, \ldots, s_k$ be a set of $0,1$ vectors and $\lambda_i \geq 0$ be such that $\sum_{i=1}^{k} \lambda_i s_i = e$, hence $\sum_i \lambda_i = t > 0$. Let $M = \sum_i \lambda_i s_i s_i^T$. Then $diag(M) = e,\ tM - J \succeq 0$.*

[2]Oberwolfach conference on Discrete Optimization, November 2008

*Proof.* Consider

$$\sum_i \lambda_i \begin{pmatrix} 1 \\ s_i \end{pmatrix} \begin{pmatrix} 1 \\ s_i \end{pmatrix}^T = \begin{pmatrix} t & e^T \\ e & M \end{pmatrix} \succeq 0.$$

We note that $diag(M) = \sum_i \lambda_i diag(s_i s_i^T) = \sum_i \lambda_i s_i = e$. The Schur-complement lemma shows that $M - \frac{1}{t} ee^T \succeq 0$. □

Proposition 3.2 clearly follows with $\lambda_i = 1$. We recall that $\delta(S_i)$ denotes the set of edges joining $S_i$ to $V \setminus S_i$. A simple calculation using basic properties of the Laplacian $L$ shows that

$$s_i^T L s_i = \sum_{uv \in \delta(S_i)} a_{uv} \tag{3.2}$$

gives the weight of all edges cut by $S_i$. Therefore the total weight of all edges joining distinct subsets is given by

$$\frac{1}{2} \sum_i s_i^T L s_i = \frac{1}{2} \langle S, LS \rangle.$$

The factor $\frac{1}{2}$ comes from the fact that an edge $uv \in E(G)$ with $u \in S_i$ and $v \in S_j$ appears in both $\delta(S_i)$ and $\delta(S_j)$. Thus Max-$k$-Cut can be modeled as

$$\max \frac{1}{2} \langle S, LS \rangle$$

such that the $n \times k$ matrix $S$ has entries 0 or 1 and $Se = e$. After replacing $SS^T$ by $Y$, we get the following SDP relaxation.

$$z_{GP-k} := \max\{\frac{1}{2}\langle L, Y \rangle : \ diag(Y) = e, \ kY - J \in \mathcal{S}^+, Y \in \mathcal{N}\} \tag{3.3}$$

The conditions $diag(Y) = e$ and $kY - J \in \mathcal{S}^+$ are derived from proposition 3.2. Note in particular that $Y \succeq 0$ is implied by $kY \succeq J \succeq 0$. The standard SDP formulation, see [19, 16], is obtained through the variable transformation

$$X = \frac{1}{k-1} [kY - J]$$

and yields, with $\langle J, L \rangle = 0$

$$\max\{\frac{k-1}{2k}\langle L, X \rangle : \ diag(X) = e, \ X \in \mathcal{S}^+, x_{ij} \geq -\frac{1}{k-1}\}. \tag{3.4}$$

**3.2. Max-Cut.** The special case of Max-$k$-Cut with $k = 2$ is usually simply called Max-Cut, as the task is to separate $V$ into $S$ and $V \setminus S$ so as to maximize the weight of edges in $\delta(S)$. In view of (3.2) we clearly have

$$z_{MC} = \max\{s^T L s : s \in \{0,1\}^n\}.$$

Setting $y = e - 2s$, we have $y \in \{1,-1\}^n$ and, using $Le = 0$, we get

$$z_{MC} = \max\{\frac{1}{4} y^T L y : \ y \in \{-1,1\}^n\}. \tag{3.5}$$

The following identity is a simple consequence of the definition of the Laplacian $L$ through the adjacency matrix $A$, see (3.1)

$$\frac{1}{4} y^T L y = \sum_{ij \in E(G)} a_{ij} \frac{1 - y_i y_j}{2}. \tag{3.6}$$

The resulting SDP relaxation becomes

$$\max\{\frac{1}{4}\langle L, X\rangle : \ diag(X) = e, \ X \in \mathcal{S}^+\}. \tag{3.7}$$

This model is identical to (3.4) with $k = 2$. We point out in particular that the sign constraint $x_{ij} \geq -1$ is implied by $X \succeq 0$ and $diag(X) = e$, and hence redundant.

**3.3. $k$-Equicut.** The following version of graph partition constrains the cardinalities of the partition blocks. In the simplest version of $k$-Equicut they are required to be equal to one another, $|S_i| = \frac{n}{k}\ \forall i$. Thus the column sums of $S$ are $\frac{n}{k}$, $S^T e = \frac{n}{k} e$. We also have $Se = e$, because each vertex is in exactly one partition block. From this it follows that $SS^T e = \frac{n}{k} e$, so $\frac{n}{k}$ is the eigenvalue of $SS^T$ for the eigenvector $e$. The relaxation from (3.3) therefore leads to

$$\min\{\frac{1}{2}\langle L, X\rangle : \ diag(X) = e, \ Xe = \frac{n}{k} e, \ X \geq 0, \ X \succeq 0\}.$$

In the context of Equicut, one is often interested in minimizing the total weight of edges cut. It is well known that without the cardinality constraints on the partition blocks, one can find the minimum cut (in the bisection case $k = 2$) using maximum flows. We also note that $X \succeq 0$ together with $Xe = \frac{n}{k} e$ implies $X = \frac{1}{k} J + \sum_i \lambda_i u_i u_i^T$ where the eigenvalues $\lambda_i \geq 0$ and the eigenvectors $u_i \perp e$. Therefore $kX - J \succeq 0$ as in (3.3) is implied. Further modeling ideas for Equicut using SDP can be found for instance in [36]. Applications of Equicut in telecommunication, and some computational experience with $k$-Equicut are shown in [41].

**3.4. Approximation results for graph partition.** The SDP relaxations for Max-Cut and Max-$k$-Cut can be used to get the following polynomial time approximations. The key idea underlying this approach was introduced by Goemans and Williamson [23] and consists of the following geometric construction. A feasible solution $X$ of (3.4) or (3.7) has the Gram representation $X = (x_{ij})$ with $x_{ij} = (v_i^T v_j)$. The constraint $diag(X) = e$ implies that the $v_i$ are unit vectors.

Let us first consider Max-Cut. Goemans and Williamson [23] interpret the Max-Cut problem as finding an embedding $v_i$ of the vertices $i$ in the unit sphere in $\mathbb{R}^1$, hence $v_i \in \{-1, 1\}$, such that $\frac{1}{4}\sum_{ij} l_{ij} v_i^T v_j$ is maximized, see (3.5).

The optimal solution of the relaxation (3.7) gives such an embedding in $\mathbb{R}^d$, where $d$ is the rank of an optimal $X$. Clearly $1 \leq d \leq n$. How should we get a (bi)partition of approximately maximum weight? Goemans and Williamson propose the following simple but powerful hyperplane rounding trick.

Take a random hyperplane $H$ through the origin, and let $S \subseteq V$ be the set of vertices on one side of $H$. The probability that $H$ separates $i$ and $j$ is proportional to the angle between $v_i$ and $v_j$ and is given by

$$\frac{1}{\pi} \arccos(v_i^T v_j).$$

In [23] it is shown that

$$\frac{1}{\pi} \arccos t \geq \alpha \frac{1}{2}(1 - t)$$

holds for $-1 \leq t \leq 1$ and $\alpha \approx 0.87856$. We therefore get the following performance bound for the expected value of the cut $y$ obtained this way, provided $a_{ij} \geq 0$.

$$\sum_{ij \in E(G)} a_{ij} \frac{1 - y_i y_j}{2} \geq \alpha \sum_{ij} a_{ij} \frac{1}{2}(1 - v_i^T v_j) \approx 0.87856 z_{MC}$$

Note the use of (3.6). Later on, Nesterov [51] generalizes this result to the more general case where only $L \succeq 0$ is assumed. The analysis in this case shows that the expected value of the cut $y$ obtained from hyperplane rounding is at least

$$\frac{1}{4} y^T L y \geq \frac{2}{\pi} z_{MC} \approx 0.636 z_{MC}.$$

Frieze and Jerrum [19] generalize the hyperplane rounding idea to Max-$k$-Cut. Starting again from the Gram representation $X = V^T V$ with unit vectors $v_i$ forming $V$, we now take $k$ independent random vectors $r_1, \ldots, r_k \in \mathbb{R}^n$ for rounding. The idea is that partition block $S_h$ contains those vertices $i$ which have $v_i$ most parallel to $r_h$,

$$i \in S_h \Longleftrightarrow v_i^T r_h = \max\{v_i^T r_l : \ 1 \leq l \leq k\}.$$

Ties are broken arbitrarily. For the computation of the probability that two vertices are in the same partition block, it is useful to assume that the entries of the $r_i$ are drawn independently from the standard normal distribution.

$$\Pr(v_s, v_t \in S_1) = \Pr(v_s^T r_1 = \max_i v_s^T r_i,\ v_t^T r_1 = \max_i v_t^T r_i).$$

The symmetry properties of the normal distribution imply that this probability depends on $\rho = \cos(v_s^T v_t)$ only. We denote the resulting probability by $I(\rho)$. Therefore

$$Pr(v_s \text{ and } v_t \text{ not separated}) = kI(\rho).$$

The computation of $I(\rho)$ involves multiple integrals. A Taylor series expansion is used in [19] to get the following estimates for the expectation value of the cut given by the partition $S$ from hyperplane rounding,

$$\frac{1}{2}\langle S, LS\rangle \geq \alpha_k z_{GP-k},$$

where $\alpha_2 = 0.87856$ as for Max-Cut, $\alpha_3 \approx 0.8327$, $\alpha_4 \approx 0.85$. In [19], values for $\alpha_k$ are also provided for larger values of $k$. Later, these bounds on $\alpha_k$ were slightly improved, see [16]. It should be emphasized that the mathematical analysis underlying this simple rounding scheme involves rather subtle techniques from classical calculus to deal with probability estimates leading to the final error bounds $\alpha_k$, see also the monograph [14].

**4. Stable sets, cliques and coloring.** The seminal work of Lovász [42] introduces a semidefinite program, which can be interpreted both as a relaxation of the Max-Clique problem and the Coloring problem.

**4.1. Stable sets and Cliques.** The Stable-Set problem has already been described in the introduction. We denote by $\alpha(G)$ the stability number of $G$ (= cardinality of the largest stable set in $G$). It is given as the optimal solution of the following integer program.

$$\alpha(G) = \max\{e^T x : x \in \{0,1\}^n,\ x_i + x_j \leq 1\ \forall ij \in E(G)\}.$$

We first observe that the inequalities could equivalently be replaced by

$$x_i x_j = 0\ \forall ij \in E(G).$$

A clique in $G$ is a subset of pairwise adjacent vertices. Moving from $G$ to the complement graph $\overline{G}$, which joins vertices $i \neq j$ whenever $ij \notin E(G)$, it is immediately clear that stable sets in $G$ are cliques in $\overline{G}$ and vice-versa.

In the spirit of matrix lifting, we introduce for a nonzero characteristic vector $x$ of some stable set the matrix

$$X := \frac{1}{x^T x} x x^T. \tag{4.1}$$

These matrices satisfy

$$X \succeq 0,\ tr(X) = 1,\ x_{ij} = 0\ \forall ij \in E(G).$$

We also have $\langle J, X\rangle = \frac{(e^T x)^2}{x^T x} = e^T x$. We collect the equations $x_{ij} = 0\ \forall ij \in E(G)$ in the operator equation $\mathcal{A}_G(X) = 0$. Therefore we get the semidefinite programming upper bound $\alpha(G) \leq \vartheta(G)$,

$$\vartheta(G) := \max\{\langle J, X\rangle :\ tr(X) = 1,\ \mathcal{A}_G(X) = 0,\ X \succeq 0\}. \tag{4.2}$$

This is in fact one of the first relaxations for a combinatorial optimization problem based on SDP. It was introduced by Lovász [42] in 1979. This problem has been the starting point for many quite far reaching theoretical investigations. It is beyond the scope of this paper to explain them in detail, but here are some key results.

Grötschel, Lovász and Schrijver [25] show that $\alpha(G)$ can be computed in polynomial time for **perfect graphs**. This is essentially a consequence of the tractability to compute $\vartheta(G)$, and the fact that $\alpha(G) = \vartheta(G)$ holds for perfect graphs $G$. We do not explain the concept of perfect graphs here, but refer for instance to [25, 60]. It is however a prominent open problem to provide a polynomial time algorithm to compute $\alpha(G)$ for perfect graphs, which is purely combinatorial (=not making use of $\vartheta(G)$).

The Stable-Set problem provides a good example for approximations based on other matrix relaxations. Looking at (4.1), we can additionally ask that $X \in \mathcal{N}$. In this case the individual equations $x_{ij} = 0\ \forall ij \in E(G)$ can be added into a single equation $\sum_{ij\in E(G)} x_{ij} = 0$. If we use $A_G$ for the adjacency matrix of $G$, this means

$$\langle A_G, X\rangle = 0.$$

Hence we get the stronger relaxation, proposed independently by Schrijver [59] and McEliece et al [47].

$$\alpha(G) \leq \max\{\langle J, X\rangle :\ \langle A_G, X\rangle = 0,\ tr(X) = 1,\ X \in \mathcal{S}^+ \cap \mathcal{N}\} =: \vartheta'(G).$$

In terms of matrix cones, we have moved from $\mathcal{S}^+$ to $\mathcal{S}^+ \cap \mathcal{N}$. We can go one step further. The matrix $X$ from (4.1) is in fact completely positive, hence we also get

$$\alpha(G) \leq \max\{\langle J, X\rangle :\ \langle A_G, X\rangle = 0,\ tr(X) = 1,\ X \in C^*\}. \tag{4.3}$$

We will see shortly that the optimal value of the copositive program on the right hand side is in fact equal to $\alpha(G)$. This result is implicitly contained in Bomze et al [8] and was stated explicitly by de Klerk and Pasechnik [15]. A simple derivation can be obtained from the following theorem of Motzkin and Straus [48]. We recall that $\Delta := \{x \in \mathbb{R}^n : x \geq 0,\ e^T x = 1\}$.

THEOREM 4.1. *[48] Let $A_G$ be the adjacency matrix of a graph $G$. Then*

$$\frac{1}{\alpha(G)} = \min\{x^T(A_G + I)x : \ x \in \Delta\}. \tag{4.4}$$

The relation (4.4) implies in particular that

$$0 = \min\{x^T(A_G + I - \frac{1}{\alpha}ee^T)x : \ x \in \Delta\} = \min\{x^T(A_G + I - \frac{1}{\alpha}J)x : \ x \geq 0\}.$$

This clearly qualifies the matrix $\alpha(A_G + I) - J$ to be copositive. Therefore

$$\inf\{\lambda : \ \lambda(A_G + I) - J \in C\} \leq \alpha(G).$$

But weak duality for conic linear programs also shows that

$$\sup\{\langle J, X\rangle : \ \langle A_G + I, X\rangle = 1, \ X \in C^*\} \leq \inf\{\lambda : \ \lambda(A_G + I) - J \in C\}.$$

Finally, any matrix $X$ of the form (4.1) is feasible for the sup-problem, hence

$$\alpha(G) \leq \sup\{\langle J, X\rangle : \ \langle A_G + I, X\rangle = 1, \ X \in C^*\}.$$

Combining the last 3 inequalities, we see that equality must hold throughout, the infimum is attained at $\lambda = \alpha(G)$ and the supremum is attained at (4.1) with $x$ being a characteristic vector of a stable set of size $\alpha(G)$. Hence we have shown the following result.

THEOREM 4.2. *[15] Let $G$ be a graph. Then*

$$\alpha(G) = \max\{\langle J, X\rangle : \ \langle A_G + I, X\rangle = 1, \ X \in C^*\}.$$

This shows on one hand that copositive programs are intractable. It also shows however, that models based on CP may be substantially stronger than SDP based models. We will see more of this in some of the subsequent sections.

**4.2. Coloring.** Partitioning the vertex set $V$ of a graph into stable sets is pictorially also called **vertex coloring**. Each partition block $S_i$ receives a distinct 'color', and vertices having the same color are non-adjacent, because $S_i$ is stable. We could for instance partition into singletons, so each vertex would get a distinct color. The **chromatic number** $\chi(G)$ is the smallest number $k$ such that $G$ has a $k$-partition into stable partition blocks.

If we let $\{S_1, \ldots\}$ denote the set of all stable sets of $G$, then the following integer program determines $\chi(G)$,

$$\chi(G) = \min\{\sum_i \lambda_i : \ \sum_i \lambda_i x_i = e, \ \lambda_i \in \{0, 1\}\}.$$

$x_i$ denotes the characteristic vector of the stable set $S_i$. It should be observed that the number of variables $\lambda_i$ is in general not polynomially bounded in $|V(G)|$. The **fractional chromatic number** $\chi_f(G)$ is obtained by allowing $\lambda_i \geq 0$,

$$\chi_f(G) = \min\{\sum_i \lambda_i : \ \sum_i \lambda_i x_i = e, \ \lambda_i \geq 0\}. \tag{4.5}$$

This is now a linear program with a possibly exponential number of variables. Computing $\chi_f(G)$ is known to be NP-hard, see for instance [60].

A semidefinite programming based lower bound on $\chi_f(G)$ can be obtained as follows, see Lovász [42] and Schrijver [60]. Let $\lambda_i \geq 0$ be an optimal solution of (4.5), so $\chi_f(G) = \sum_i \lambda_i$. Since $\sum_i \lambda_i x_i = e$, we can apply Lemma 3.1. The matrix

$$M = \sum_i \lambda_i x_i x_i^T \tag{4.6}$$

therefore satisfies

$$\chi_f(G)M - J \in \mathcal{S}^+, \ diag(M) = e.$$

Moreover, since each $x_i$ is characteristic vector of a stable set in $G$, we also have $m_{uv} = 0$ $uv \in E(G)$, or $\mathcal{A}_G(M) = 0$. Therefore the optimal value of the following SDP is a lower bound on $\chi_f(G)$,

$$\chi_f(G) \geq \min\{t : \ diag(M) = e, \ \mathcal{A}_G(M) = 0, tM - J \succeq 0\}. \tag{4.7}$$

Strictly speaking, this is not a linear SDP, because both $t$ and $M$ are variables, but it can easily be linearized by introducing a new matrix variable $Y$ for $tM$ and asking that $diag(Y) = te$. The resulting problem is the dual of

$$\max\{\langle J, X\rangle : \ \langle I, X\rangle = 1, \ x_{ij} = 0 \ ij \notin E(G), \ X \succeq 0\},$$

which is equal to $\vartheta(\overline{G})$. Thus we have shown the Lovász 'sandwich theorem'. In [42], the weaker upper bound $\chi(G)$ is shown for $\vartheta(\overline{G})$, but it is quite clear that the argument goes through also with $\chi_f(G)$.

THEOREM 4.3. *[42] Let $G$ be a graph. Then $\alpha(\overline{G}) \leq \vartheta(\overline{G}) \leq \chi_f(G)$.*

Let us now imitate the steps leading from $\vartheta(G)$ to the copositive strengthening of the stability number from the previous section. The crucial observation is that $M$ from (4.6) and therefore $tM \in C^*$.

This leads to the following conic problem, involving matrices both in $\mathcal{S}^+$ and $C^*$.

$$t^* := \min\{t : \ diag(M) = e, \ \mathcal{A}_G(M) = 0, tM - J \in \mathcal{S}^+, \ M \in C^*\}.$$

Using again Lemma 3.1 we see that $M$ from (4.6) is feasible for this problem, therefore

$$t^* \leq \chi_f(G).$$

In [18] it is in fact shown that equality holds.

THEOREM 4.4. *[18] The optimal value $t^*$ of the above SDP-CP relaxation of the chromatic number is equal to $\chi_f(G)$.*

This shows again the strength of modeling with copositive programs. New relaxations of the chromatic number based on graph products have very recently been introduced in [26, 27, 28]. It is beyond the scope of this introductory survey to elaborate on this approach.

**4.3. Approximation results for Coloring.** Similar to Max-$k$-Cut we can use the SDP relaxations of coloring to derive a vertex partition using hyperplane rounding. An additional complication comes from the fact that the partition blocks have to be stable sets. The first groundbreaking results were obtained by Karger et al [35]. We briefly explain some of the ideas for the case of graphs $G$ having $\chi(G) = 3$. This may seem artificial, but simply knowing that $\chi(G) = 3$ does not help much. In fact, finding a 4-coloring in a 3-colorable graph is NP-hard, see [37].

Widgerson [68] observes that if the largest degree $d_{\max}$ of a graph on $n$ vertices is large, $d_{\max} > \sqrt{n}$, then 2 colors suffice to color the neighbourhood of the vertex with largest degree, thereby legally coloring at least $\sqrt{n}$ vertices. If $d_{\max} \leq \sqrt{n}$, then the graph can be colored with $\sqrt{n}+1$ colors. This yields a simple algorithm that colors any three-colorable graph with at most $3\sqrt{n}$ colors, see [68].

In [35], the SDP underlying $\vartheta(G)$ is used for hyperplane rounding. The key observation is that a carefully chosen rounding procedure can be used to produce with high probability a stable set of size $\tilde{O}(\frac{n}{d_{\max}^{1/3}})$. The $\tilde{O}$ notation ignores polylogarithmic terms. This leads to a coloring with $\tilde{O}(n^{1/4})$ colors. Based on this approach, Blum and Karger [6] refine the analysis and end up with colorings using $\tilde{O}(n^{3/14})$ colors. Note that $3/14 \approx 0.2143$. Very recently, these results were further improved by Arora et al [2] to at most $\tilde{O}(n^{0.2111})$ colors. The derivation of these estimates is rather complex. We refer to the recent dissertation [13] for a detailed discussion of hyperplane rounding for coloring.

**5. Bandwidth of graphs.** The **minimum bandwidth problem** of a graph $G$ can be interpreted as a reordering problem of the vertices of $G$ such that the nonzero entries of the adjacency matrix (after reordering) are collected within a band of small width around the main diagonal. Formally we denote the vertices of $G$ again by $V = \{1, \ldots, n\}$. For a permutation $\phi \in \Pi$, the bandwidth $bw(\phi)$ of $\phi$ is defined as

$$bw(\phi) := \max\{|\phi(i) - \phi(j)| : \ ij \in E(G)\}.$$

The bandwidth of $G$ is the smallest of these values over all bijections $\phi : V \mapsto V$.

$$bw(G) := \min\{bw(\phi) : \ \phi \in \Pi\}. \tag{5.1}$$

Determining $bw(G)$ is in general NP-hard, see [52]. It remains NP-hard, even if $G$ is restricted to be a tree with maximum degree 3, see [45].

The bandwidth problem has many applications. Consider for instance a sparse symmetric system of linear equations. Having an ordering of the system matrix with small bandwidth may result in a substantial computational speedup, when actually solving the system.

In the following approximation approach, Blum et al [7] formulate the bandwidth problem as an ordering of $V$ on $n$ equidistant points on a quarter-circle of radius $n$. Let

$$P_j := n(\cos\frac{j\pi}{2n}, \ \sin\frac{j\pi}{2n}), \ j = 1, \ldots, n$$

denote these points. The problem

$$\min\{b : \ \exists\phi \in \Pi \text{ such that } v_i = P_{\phi(i)}, \ \|v_i - v_j\| \leq b \ \forall ij \in E(G)\}$$

is clearly equivalent to finding $bw(G)$. Blum et al relax the difficult part of bijectively mapping $V$ to $\{P_1, \ldots, P_n\}$. Let us define the constraints

$$\|v_i\| = n \ \forall i \in V, \ \|v_i - v_j\| \leq b \ \forall ij \in E. \tag{5.2}$$

Simply solving

$$\min\{b : \ v_i \text{ satisfy (5.2)}\}$$

could be done using SDP by introducing $X = V^T V \succeq 0$. The constraints (5.2) translate into

$$x_{ii} = n^2 \ \forall i, \ x_{ii} + x_{jj} - 2x_{ij} \leq \beta \ \forall ij \in E \tag{5.3}$$

and minimizing $\beta$ would yield $b = \sqrt{\beta}$. Unfortunately, the optimum of this SDP has value $\beta = 0$, by assigning each $i$ to the same point, say $P_1$. To force some spreading of the $v_i$, we observe the following.

PROPOSITION 5.1. *Let $i \in V$ and $S \subseteq V \setminus i$. Then*

$$\sum_{j \in S}(i-j)^2 \geq \frac{|S|}{6}(|S|+1)(\frac{|S|}{2}+1) =: f(|S|).$$

Since $\|P_i - P_j\| \geq |i-j|$, we can include these additional spread constraints

$$\sum_{j \in S}\|v_i - v_j\|^2 \geq f(|S|) \ \forall S \subseteq V, \ i \in V \tag{5.4}$$

into the SDP. Blum et al [7] consider the following strenghtened problem, which is equivalent to an SDP, once we make the change of variables $X = V^T V$.

$$\min\{b : \ (v_i) \text{ satisfy } (5.2), (5.4)\}. \tag{5.5}$$

Even though there is an exponential number of these spread constraints, it can easily be argued that their separation can be done in polynomial time by sorting, for $i$ fixed, the remaining vertices $j \in V \setminus i$ in increasing order of $\|v_i - v_j\|^2$.

Having an optimal solution $b, v_1, \ldots, v_n$ of the SDP, Blum et al apply the hyperplane rounding idea as follows. Take a random line through the origin and project the $v_i$ onto this line. The resulting permutation of the vertices is shown to satisfy the following estimate.

THEOREM 5.1. *[7] Let $G$ be a graph and $b, v_1, \ldots, v_n$ denote the optimal solution of (5.5). The ordering $\phi$ produced by projecting the $v_i$ onto a random line through the origin satisfies*

$$bw(\phi) \leq O(\sqrt{n/b}\log(n))bw(G).$$

*with high probability.*

The proof of this result is rather long and technical and is omitted here.

We are now going to describe another modeling approach which can be used to bound $bw(G)$. Let $G$ be a connected graph, given through its adjacency matrix $A \geq 0$. We now consider 3-partitions $(S_1, S_2, S_3)$ of $V$ having prescribed cardinalities $|S_i| = m_i$. Let us denote the edges joining $S_1$ and $S_2$ by $\delta(S_1, S_2)$. If we set $B = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$ and use the characteristic vectors $s_i$ for $S_i$, we get the total weight of edges in $\delta(S_1, S_2)$ by

$$s_1^T A s_2 = \frac{1}{2}\langle S, ASB\rangle.$$

Minimizing this cost function over 3-partitions of prescribed cardinalities $m = (m_1, m_2, m_3)$ may look artificial, but we will see that it provides a handle to several graph optimization problems, in particular the bandwidth problem. So let us consider

$$z_{3gp} := \min\{\frac{1}{2}\langle S, ASB\rangle : \ Se = e, S^T e = m, S \geq 0, \ S^T S = M\}. \tag{5.6}$$

We set $M = Diag(m)$. This is a nonconvex quadratic optimization problem, hence intractable. We leave it to the reader to verify that feasible

matrices $S$ of this problem are integer. We can use this problem as follows. If $z_{3gp} = 0$ then the graph underlying $A$ has a **vertex separator** of size $m_3$. (We recall that $S \subset V$ is a vertex separator, if the removal of $S$ disconnects the graph.) If $z_{3gp} > 0$ then the bandwidth of $A$ is at least $m_3 + 1$, see Helmberg et al [31]. This last condition could also be used if we would have a positive lower bound on $z_{3gp}$. In [31], such a lower bound was derived using eigenvalue techniques on the basis that the columns of $S$ are pairwise orthogonal. Povh and Rendl [54] consider the matrix lifting

$$\mathcal{M}_{3gp} := conv\{ss^T : s = vec(S), S \text{ is 3-partition with } |S_i| = m_i\}. \tag{5.7}$$

It turns out that $\mathcal{M}_{3gp}$ can be represented as the intersection of a set of linear equations with the cone $C^*$ of completely positive matrices. To see how these equations are derived, we start out with the quadratic equation $S^T S = M$. At this point the Kronecker product of two matrices $P$ and $Q$, given by $P \otimes Q := (p_{ij}Q)$, is extremely useful. If $P, Q$ and $S$ have suitable size, and $s = vec(S)$, the following identity is easy to verify.

$$\langle S, PSQ\rangle = \langle Q^T \otimes P, ss^T\rangle. \tag{5.8}$$

Using it, we see that $(S^T S)_{ij} = e_i^T S^T S e_j = \langle e_j e_i^T \otimes I, ss^T\rangle$. The symmetry of $ss^T$ allows us to replace $e_j e_i^T$ by $B_{ij} := \frac{1}{2}(e_j e_i^T + e_i e_j^T)$. Therefore $(S^T S)_{ij} = M_{ij}$ becomes

$$\langle B_{ij} \otimes I, Y\rangle = M_{ij} \tag{5.9}$$

where $Y \in \mathcal{M}_{3gp}$. In a similar way we get

$$\langle J_3 \otimes e_i e_i^T, Y\rangle = 1,\ 1 \le i \le n, \tag{5.10}$$

by squaring the $n$ equations $Se = e$. Pairwise multiplication of the constraints $S^T e = m$ gives

$$\langle B_{ij} \otimes J_n, Y\rangle = m_i m_j\ 1 \le i \le j \le 3. \tag{5.11}$$

Finally, elementwise multiplication of $Se = e$ with $S^T e = m$ gives

$$\langle e_i e^T \otimes e e_j^T, Y\rangle = m_i\ 1 \le i \le 3,\ 1 \le j \le n. \tag{5.12}$$

The following result is shown in [54].

THEOREM 5.2.

$$\mathcal{M}_{3gp} = \{Y : Y \in C^*,\ Y \textit{ satisfies } (5.9), (5.10), (5.11), (5.12)\}.$$

Therefore $z_{3gp}$ could be determined as the optimal solution of the (intractable) copositive program

$$z_{3gp} = \min\{\frac{1}{2}\langle B \otimes A, Y\rangle :\ Y \in C^*,\ \ Y \text{ satisfies } (5.9), (5.10), (5.11), (5.12)\}.$$

We get a tractable relaxation by asking $Y \in \mathcal{S}^+$ instead of $Y \in C^*$. Further details can be found in [54] and in the dissertation [53]. While the model investigated by [7] is based on $n \times n$ matrices, it has to be emphasized that the last model uses $3n \times 3n$ matrices, and hence is computationally significantly more demanding.

**6. Quadratic assignments.** The quadratic assignment problem, or QAP for short, is a generalization of the (linear) assignment problem, briefly described in the introduction. It consists of minimizing a quadratic function over the set of permutation matrices $\{X_\phi : \ \phi \in \Pi\}$. The general form of QAP, for given symmetric $n^2 \times n^2$ matrix $Q$ is as follows

$$z_{QAP} := \min\{x_\phi^T Q x_\phi : \ x_\phi = vec(X_\phi), \ \phi \in \Pi\}.$$

In applications, $Q$ is often of the form $Q = B \otimes A$, where $A$ and $B$ are symmetric $n \times n$ matrices. In this case the objective function has a representation in terms of $n \times n$ matrices, see (5.8)

$$x_\phi^T (B \otimes A) x_\phi = \langle X_\phi, A X_\phi B \rangle.$$

We refer to the recent monograph [12] by Burkard et al for further details on assignment problems.

To get a handle on QAP, we consider the matrix lifting

$$\mathcal{M}_{QAP} := conv\{x_\phi x_\phi^T : x_\phi = vec(X_\phi), \ \phi \in \Pi\}.$$

We will now see that $\mathcal{M}_{QAP}$ has a 'simple' description by linear equations intersected with the cone $C^*$. Matrices $Y \in \mathcal{M}_{QAP}$ are of order $n^2 \times n^2$. It will be useful to consider the following partitioning of $Y$ into $n \times n$ block matrices $Y^{i,j}$,

$$Y = \begin{pmatrix} Y^{1,1} & \dots & Y^{1,n} \\ \vdots & \ddots & \vdots \\ Y^{n,1} & \dots & Y^{n,n} \end{pmatrix}.$$

Let $X = (x_1, \dots, x_n)$ be a permutation matrix (with columns $x_i$). Then $X$ can be characterized by $X \geq 0, X^T X = X X^T = I$. These quadratic constraints translate into linear constraints on $Y$:

$$X X^T = \sum_i x_i x_i^T = \sum_i Y^{i,i} = I.$$

Similarly, $(X^T X)_{ij} = x_i^T x_j = tr(x_i x_j^T) = tr(Y^{i,j}) = \delta_{ij}$. Finally, we have $(\sum_{ij} x_{ij})^2 = n^2$ for any permutation matrix $X$. We get the following set of constraints for $Y$:

$$\sum_i Y^{i,i} = I, \ tr(Y^{i,j}) = \delta_{ij}, \ \langle J, Y \rangle = n^2. \tag{6.1}$$

Povh and Rendl [55] show the following characterization of $\mathcal{M}_{QAP}$, which can be viewed as a lifted version of Birkhoff's theorem 1.1.

THEOREM 6.1. $\mathcal{M}_{QAP} = \{Y : \ Y \in C^*, \ Y \ \textit{satisfies} \ (6.1)\}$.

It is not hard to verify that the above result would be wrong without the seemingly redundant equation $\langle J, Y\rangle = n^2$. We can therefore formulate the quadratic problem QAP as a (linear but intractable) copositive program

$$z_{QAP} = \min\{\langle Q, Y\rangle : \ Y \in C^*, \ Y \text{ satisfies } (6.1)\}.$$

In [55] some semidefinite relaxations based on this model are investigated and compared to previously published SDP relaxations of QAP.

We have now seen several instances of combinatorial optimization problems, where CP relaxations in fact gave the exact value. This raises the question whether there is some general principle behind this observation. Burer [10] gives a rather general answer and shows that an appropriate reformulation of quadratic programs is equivalent to a linear copositive program.

THEOREM 6.2. *[10] Let $c$ and $a_j$ be vectors from $\mathbb{R}^n$, $b \in \mathbb{R}^k$, $Q \in \mathcal{S}_n$ and $I \subseteq \{1, \ldots, n\}$. The optimal values of the following two problems are equal.*

$$\min\{x^T Q x + c^T x : a_j^T x = b_j, \ x \geq 0, \ x_i \in \{0, 1\} \ i \in I\},$$

$$\min\{\langle Q, X\rangle + c^T x : a_j^T x = b_j, a_j^T X a_j = b_j^2,$$

$$X_{ii} = x_i \ \forall i \in I, \ \begin{pmatrix} 1 & x^T \\ x & X \end{pmatrix} \in \mathcal{C}^*\}.$$

**7. Optimal mixing rate of Markov chains.** In the previous sections we focused on various ways to get matrix liftings of NP-hard optimization problems. We conclude now with an SDP model which optimizes the mixing rate of a finite Markov chain. Let $G$ be a connected graph. We consider random walks on the vertex set $V(G)$. Suppose we can either stay at $i \in V(G)$ or move to $j$, provided $ij \in E(G)$. Suppose further that the transition probabilities $p_{ij}$ are symmetric, $p_{ij} = p_{ji} \ \forall ij$. The resulting Markov chain is now described by the transition matrix $P$ satisfying

$$P \in \mathcal{P}_G := \{P : P = P^T, \ P \in \Omega, \ p_{ij} = 0 \ ij \notin E(G)\}.$$

Finally, we assume that $P$ is primitive (aperiodic and irreducible). This means there exists some $k$ such that $P^k > 0$. Let us first recall the Perron-Frobenius theorem for primitive matrices.

THEOREM 7.1. *Let $P$ be a nonnegative primitive square matrix. Then the spectral radius is a simple eigenvalue of $P$ with eigenvector $x > 0$.*

We denote by $\pi(t) \in \mathbb{R}^n$ the probability distribution on $V$ at time $t$. The definition of the transition probabilities in $P$ imply that $\pi(t+1) = P^T\pi(t)$. Symmetry of $P$ therefore shows that $\pi(t)$ is determined from the initial distribution $\pi(0)$ through $\pi(t) = P^t\pi(0)$. We denote the eigenvalues of $P$ by

$$1 = \lambda_1(P) > \lambda_2(P) \geq \ldots \geq \lambda_n(P) > -1.$$

The Perron-Frobenius theorem tells us that $\frac{1}{n}e$ is eigenvector to the eigenvalue 1, which is also the spectral radius of $P$. Therefore

$$\lim_{t\to\infty} \pi(t) = \frac{1}{n}e.$$

What can be said about the speed of convergence? There are several ways to measure the distance of $\pi(t)$ from the equilibrium distribution $\pi(\infty) = \pi = \frac{1}{n}e$. One such measure is the maximum relative error at time $t$

$$r(t) := \max_{ij} \frac{|(P^t)_{ij} - \pi_j|}{\pi_j},$$

see for instance [3]. Let

$$\mu_P := \max\{\lambda_2(P), -\lambda_n(P)\} = \max\{|\lambda_i(P)| : \ i > 1\}$$

denote the second largest eigenvalue of $P$ in modulus (SLEM). It is well known that $\mu_P$ is closely related to how fast $\pi(t)$ converges to the equilibrium distribution $\frac{1}{n}e$.

THEOREM 7.2. *[3] Let $P$ be a symmetric irreducible transition matrix. Then*

$$r(t) \leq n(\mu_P)^t.$$

*Moreover $r(t) \geq (\mu_P)^t$ if $t$ is even.*

Given the graph $G$, we can ask the question to select the transition probabilities $p_{ij} > 0$ for $ij \in E(G)$ in such a way that the mixing rate of the Markov chain is as fast as possible. In view of the bounds from the previous theorem, it makes sense to consider the following optimization problem (in the matrix variable $P$), see Boyd et al [9].

$$\min\{\mu_P : P \in \mathcal{P}_G\}.$$

They show that $\mu_P$ is in fact a convex function. It follows from the Perron-Frobenius theorem and the spectral decomposition theorem for symmetric matrices that $\mu_P$ is either the smallest or the largest eigenvalue of $P - \frac{1}{n}J$ in absolute value. Hence we can determine $\mu_P$ as the solution of the following SDP, see [9].

$$\min\{s : \ sI \succeq P - \frac{1}{n}J \succeq -sI, \ P \in \mathcal{P}_G\} \tag{7.1}$$

The variables are $s$ and the matrix $P$. In [9], it is shown that an optimal choice of $P$ may significantly increase the mixing rate of the resulting chain. Some further extensions of this idea are discussed in [64].

**8. Computational progress.** Up to now we have mostly concentrated on the modeling power of SDP and CP. From a practical point of view, it is also important to investigate the algorithmic possibilities to actually solve the relaxations. Solving copositive programs is at least as hard as general integer programming, hence we only consider solving SDP, which are tractable. Before we describe various algorithms to solve SDP, we recall the basic assumptions from Theorem 2.1,

$$\exists X_0 \succ 0,\ y_0 \text{ such that } A(X) = b,\ C - A^T(y_0) \succ 0. \tag{8.1}$$

In this case $(X, y, Z)$ is optimal for the primal-dual pair

$$\min\{\langle C, X\rangle :\ A(X) = b,\ X \succeq 0\} = \max\{b^T y :\ C - A^T(y) = Z \succeq 0\}$$

if and only if

$$A(X) = b,\ A^T(y) + Z = C,\ X \succeq 0,\ Z \succeq 0,\ \langle X, Z\rangle = 0. \tag{8.2}$$

We are now going to briefly describe several classes of algorithms to solve SDP and point out their strengths and limitations.

**8.1. Interior-point methods.** There exist several quite in-depth descriptions of primal-dual interior-point path-following methods for SDP. We refer to [67, 63, 14] and the SDP handbook [70]. The website [3] maintains a collection of software, various data sets and provides benchmark comparisons of several competing software packages to solve SDP.

We therefore explain here only the basic ideas. First, it follows from $X \succeq 0,\ Z \succeq 0,\ \langle X, Z\rangle = 0$ that in fact $XZ = 0$. Interior-point (path-following) methods can be viewed as a sequence of problems parametrized by $\mu \geq 0$. Consider the set $P_\mu$, defined as follows:

$$P_\mu := \{(X, y, Z) :\ A(X) = b,\ Z + A^T(y) = C,\ X \succeq 0, Z \succeq 0, ZX = \mu I\}.$$

Clearly, $P_0 \neq \emptyset$, if (8.1) holds. It can in fact be shown that $P_\mu$ defines a unique point $(X_\mu, y_\mu, Z_\mu)$ for any $\mu > 0$ if and only if (8.1) holds. See for instance Theorem 10.2.1 in [70]. In this case the set $\{(X_\mu, y_\mu, Z_\mu) : \mu \geq 0\}$ defines a smooth curve, parametrized by $\mu > 0$. Following this path until $\mu \approx 0$ clearly leads to an optimal solution of SDP. This is the basic idea underlying interior-point methods. There is quite a variety of different approaches to achieve this goal. Todd [65] gives a detailed summary of popular variants to solve SDP by path-following methods.

[3] http://plato.asu.edu/bench.html

The crucial step in all these variants consists of the following. Given a current iterate $(X_k, y_k, Z_k)$ with $X_k \succ 0$ and $Z_k \succ 0$ and a target path parameter $\mu_k > 0$, we use the Newton method to determine a search direction $(\Delta X, \Delta y, \Delta Z)$ towards (the point) $P_{\mu_k}$. If there are $m$ equations in the primal problem, so $b \in \mathbb{R}^m$, this amounts to setting up and solving a (dense) linear system of order $m$ to determine $\Delta y$. To set up this system, and to recover $\Delta X$ and $\Delta Z$, some additional computational effort is necessary, involving matrix operations (multiplication, inversion) with matrices of order $n$. Having the search direction, one needs to test whether the full Newton step is feasible ($X_k + \Delta X \succ 0$ and $Z_k + \Delta Z \succ 0$). If not, some sort of backtracking strategy is used to find a smaller steplength leading to a new iterate in the interior of $\mathcal{S}^+$. Then a new (smaller) target value for $\mu$ is selected and the process is iterated until $\mu \approx 0$ and the current iterates are primal and dual feasible.

The convergence analysis shows that under suitable parameter settings it takes $O(\sqrt{n})$ Newton iterations to reach a solution with the required accuracy. Typically, the number of such iterations is not too large, often only a few dozen, but both the memory requirements (a dense $m \times m$ matrix has to be handled) and the computation times grow rapidly with $n$ and $m$. To give some impression, we provide in Table 1 some sample timings to solve the basic relaxation for Max-Cut, see (3.7). It has $m = n$ rather simple constraints $x_{ii} = 1$. We also consider computing the theta number $\vartheta(G)$ (4.2), see Table 2. Here the computational effort is also influenced by the cardinality $|E(G)|$. We consider dense graphs ($m = \frac{1}{2}\binom{n}{2}$) and sparse graphs ($m = 5n$). In the first case, the number $n$ of vertices can not be much larger than about 200, in the second case we can go to much larger graphs. Looking at these timings, it is quite clear that interior-point methods will become impractical once $n \approx 3000$ or $m \approx 5000$.

There have been attempts to overcome working explicitly with the dense system matrix of order $m$. Toh [66] for instance reports quite encouraging results for larger problems by iteratively solving the linear system for the search direction. A principal drawback of this approach lies in the fact that the system matrix gets ill-conditioned, as one gets close to the optimum. This implies that high accuracy is not easily reachable. We also mention the approach from Kocvara and Stingl [38], which uses a modified 'barrier function' and also handles large-scale problems. Another line of research to overcome some of these limitations consists in exploiting sparsity in the data. We refer to [20, 50] for some first fundamental steps in this direction.

**8.2. Projection methods.** To overcome some of the computational bottlenecks of interior-point methods, we can exploit the fact that the projection of an arbitrary symmetric matrix $M$ to the cone of semidefinite matrices can be obtained through a spectral decomposition of $M$. More precisely, let $M = \sum_i \lambda_i u_i u_i^T$ with pairwise orthogonal eigenvectors $u_i$.

| $n$ | time (secs.) |
|---|---|
| 1000 | 12 |
| 2000 | 102 |
| 3000 | 340 |
| 4000 | 782 |
| 5000 | 1570 |

TABLE 1

*Interior-point computation times to solve (3.7) with relative accuracy* $10^{-6}$. *Here* $m = n$.

| $n$ | $m = \frac{1}{2}\binom{n}{2}$ | time (secs.) | $n$ | $m = 5n$ | time (secs.) |
|---|---|---|---|---|---|
| 100 | 2488 | 12 | 500 | 2500 | 14 |
| 150 | 5542 | 125 | 1000 | 5000 | 120 |
| 200 | 9912 | 600 | 1500 | 7500 | 410 |

TABLE 2

*Interior-point computation times to solve (4.2) with relative accuracy* $10^{-6}$, $m = \frac{1}{2}\binom{n}{2}$ *and* $m = 5n$.

Then

$$\operatorname{argmin}\{\|M - X\| : X \succeq 0\} = \sum_{i:\lambda_i > 0} \lambda_i u_i u_i^T =: M^+, \tag{8.3}$$

see for instance [24].

A rather natural use of projection was recently proposed in [34] and can be explained as follows. We recall the optimality conditions (8.2) and observe that $\langle X, Z\rangle = 0$ can be replaced by the linear equation $b^T y - \langle C, X\rangle = 0$. Hence we can group the optimality conditions into the affine linear constraints

$$(L_P)\ \ A(X) = b, \quad (L_D)\ \ A^T(y) + Z = C, \quad (L_C)\ \ \langle C, X\rangle - b^T y = 0,$$

and the SDP conditions $X \succeq 0,\ \ Z \succeq 0$. The projection onto SDP is given by (8.3). Projecting onto an affine space is also quite easy. Linear algebra tells us that the projection $\Pi_P(X)$ of a symmetric matrix $X$ onto $(L_P)$ is given by

$$\Pi_P(X) := X - A^T(AA^T)^{-1}(A(X) - b),$$

and similarly, $Z$ has the projection

$$\Pi_D(Z) := C + A^T(AA^T)^{-1}A(Z - C)$$

onto $L_D$. Thus $\Pi_D(Z) = C + A^T(y)$ with $y = (AA^T)^{-1}A(Z - C)$. Finally, the projection onto the hyperplane $L_C$ is trivial. Thus one can use alternate projections to solve SDP. Take a starting point $(X, y, Z)$, and project

it onto the affine constraints. This involves solving two linear equations with system matrix $AA^T$, which remains unchanged throughout. Then project the result onto the SDP cone and iterate. This requires the spectral decomposition of both $X$ and $Z$.

This simple iterative scheme is known to converge slowly. In [34] some acceleration strategies are discussed and computational results with $m \approx 100,000$ are reported.

Another solution approach for SDP using only SDP projection and solving a linear equation with system matrix $AA^T$ is proposed by Povh et al [56] and Malick et al [46]. The approach from [34] can be viewed as maintaining $A(X) = b,\ Z + A^T(y) = C$ and the zero duality gap condition $b^T y = \langle C, X\rangle$ and trying to get $X$ and $Z$ into $S^+$. In contrast, the approach from [56, 46] maintains $X \succeq 0,\ Z \succeq 0, ZX = 0$ and tries to reach feasibility with respect to the linear equations. The starting point of this approach consists of looking at the augmented Lagrangian formulation of the dual SDP. Let

$$f_{\sigma,X}(y,Z) := b^T y - \langle X, A^T(y) + Z - C\rangle - \frac{\sigma}{2}\|A^T(y) + Z - C\|^2 \tag{8.4}$$

and consider

$$\max_{y,Z\succeq 0} f_{\sigma,X}(y,Z).$$

Having (approximate) maximizers $y, Z$ (for $\sigma$ and $X$ held constant), the augmented Lagrangian method, see [5], asks to update $X$ by

$$X \leftarrow X + \sigma(Z + A^T(y) - C) \tag{8.5}$$

and iterate until dual feasibility is reached. The special structure of the subproblem given by (8.4) allows us to interpret the update (8.5) differently. After introducing the Lagrangian $L = f_{\sigma,X}(y,Z) + \langle V, Z\rangle$ with respect to the constraint $Z \succeq 0$, we get the following optimality conditions for maximizing (8.4).

$$\nabla_y L = b - A(X) - \sigma A(A^T(y) + Z - C) = 0,$$

$$\nabla_Z L = V - X - \sigma(A^T(y) + Z - C) = 0,\ V \succeq 0,\ Z \succeq 0,\ \langle V, Z\rangle = 0.$$

The condition $\nabla_Z L = 0$ suggests to set $X = V$, see (8.5). We note that $L$ could also be written as

$$L = b^T y - \frac{\sigma}{2}\|Z - (C - A^T(y) - \frac{1}{\sigma}X)\|^2 + \frac{1}{2\sigma}\|X\|^2.$$

Therefore, at the optimum, $Z$ must also be the projection of $W := C - A^T(y) - \frac{1}{\sigma}X$ onto $\mathcal{S}^+$, $Z = W^+$. Thus $\nabla_Z L = 0$ becomes

$$V = \sigma(Z + \frac{1}{\sigma}X + A^T(y) - C) = \sigma(W^+ - W) = -\sigma W^-,$$

| $n$ | $m = \frac{1}{4}n^2$ | time (secs.) |
|---|---|---|
| 400 | 40000 | 40 |
| 600 | 90000 | 100 |
| 800 | 160000 | 235 |
| 1000 | 250000 | 530 |
| 1200 | 360000 | 1140 |

TABLE 3
*Boundary-point computation times to solve (4.2) with relative accuracy* $10^{-6}$.

where $W^-$ is the projection of $W$ onto $-(\mathcal{S}^+)$. This leads to the boundary point method from [56]. Given $X, Z$, solve the $\nabla_y L = 0$ for $y$:

$$AA^T(y) = \frac{1}{\sigma}(b - A(X) - A(Z - C)).$$

Then compute the spectral decomposition of $W = C - A^T(y) - \frac{1}{\sigma}X$ and get a new iterate $X = -\sigma W^-, Z = W^+$ and iterate.

The computational effort of one iteration is essentially solving the linear system with matrix $AA^T$ and computing the factorization of $W$. Further details, like convergence analysis and parameter updates are described in [56, 46]. To show the potential of this approach, we compute $\vartheta(G)$ for larger dense graphs with $m = \frac{1}{4}n^2$, see Table 3. It is clear from these results that projection methods extend the computational possibilities for SDP. Zhao et al [71] recently generalized this approach to include second order information in the updates. At higher computational cost they get more accurate solutions.

The computational limitations of projection methods are determined on one hand by the need to compute a spectral decomposition of a symmetric matrix, which limits the matrix dimension similar to interior-point methods. On the other hand, the system matrix $AA^T$ does not change during the algorithm, and any sparsity properties of the input can therefore be fully exploited. In case of computing the $\vartheta$ function for instance, it turns out that $AA^T$ is in fact diagonal. This is one explanation why instances with $m$ beyond 100,000 are easily manageable by projection methods.

**8.3. Further solution approaches for SDP.** To avoid limits on the size of primal matrices (given through the spectral decomposition or the linear algebra of interior-point methods), one can also use the encoding of semidefiniteness through the condition $\lambda_{\min}(X) \geq 0$. The computation of $\lambda_{\min}(X)$, for symmetric $X$, can be done iteratively for quite large matrices. The only requirement is that $v = Xu$ can be evaluated. In particular, $X$ need not be stored explicitly. The **spectral bundle method** exploits this fact, and applies to SDP, where (primal) feasibility implies constant trace. It was introduced in [32] and we refer to this paper for all further details. It turns out that SDP with rather large matrices ($n \approx 10,000$) can be handled

by this method, but the convergence properties are much weaker than in the case of interior-point methods. Helmberg [30] describes computational results on a variety of large scale combinatorial optimization problems.

Another idea to get rid of the semidefiniteness condition is to use the the factorization $X = RR^T$, and work with $R$ using algorithms from nonlinear optimization. Burer and Monteiro [11] investigate this approach and present some encouraging computational results for some specially structured SDP like the Max-Cut relaxation (3.7). The drawback here is that by going from $X$ to the factor $R$, one loses convexity.

Finally, SDP based relaxations have been used successfully to get exact solutions. Exact solutions of the Max-Cut problem are reported in [58] for instances having several hundred vertices, see also the thesis [69]. A combination of polyhedral and SDP relaxations for the bisection problem is studied in [1]. Exact solutions of rather large sparse instances ($n \approx 1000$) are obtained for the first time. Finally, exact solutions for Max-$k$-Cut are given in [21].

Algorithms for linear optimization have reached a high level of sophistication, making them easily accessible even for non-experts. In contrast, most algorithms for SDP require a basic understanding of semidefinite optimization. While small problems $n \approx 100$ can be solved routinely, there is no standard way to solve medium sized SDP in a routine way. Finally, it is a challenging open problem to find efficient ways of optimizing over $\mathcal{S}^+ \cap \mathcal{N}$.

**Acknowledgement:** Many thanks go to Nathan Krislock and an anonymous referee for giving numerous suggestions for improvement, and for pointing out minor inconsistencies.

## REFERENCES

[1] M. Armbruster, M. Fügenschuh, C. Helmberg, and A. Martin. A comparative study of linear and semidefinite branch-and-cut methods for solving the minimum graph bisection problem. In *Integer programming and combinatorial Optimization (IPCO 2008)*, pages 112–124, 2008.

[2] S. Arora, E. Chlamtac, and M. Charikar. New approximation guarantee for chromatic number. In *Proceedings of the 38th STOC, Seattle, USA*, pages 215–224, 2006.

[3] E. Behrends. *Introduction to Markov chains with special emphasis on rapid mixing.* Vieweg Advanced Lectures in Mathematics, 2000.

[4] A. Berman and N. Shaked-Monderer. *Completely positive matrices.* World Scientific Publishing, 2003.

[5] D.P. Bertsekas. *Constrained optimization and Lagrange multipliers.* Academic Press, 1982.

[6] A. Blum and D. Karger. An $\tilde{O}(n^{3/14})$-coloring algorithm for 3-colorable graphs. *Information processing letters*, 61(1):249–53, 1997.

[7] A. Blum, G. Konjevod, R. Ravi, and S. Vempala. Semi-definite relaxations for minimum bandwidth and other vertex-ordering problems. *Theoretical Computer Science*, 235(1):25–42, 2000.

[8] I.M. Bomze, M. Dür, E. de Klerk, C. Roos, A.J. Quist, and T. Terlaky. On copos-

itive programming and standard quadratic optimization problems. *Journal of Global Optimization*, 18(4):301–320, 2000.

[9] S. Boyd, P. Diaconis, and L. Xiao. Fastest mixing Markov chain on a graph. *SIAM Review*, 46(4):667–689, 2004.

[10] S. Burer. On the copositive representation of binary and continuous nonconvex quadratic programs. *Mathematical Programming (A)*, 120:479–495, 2009.

[11] S. Burer and R.D.C. Monteiro. A nonlinear programming algorithm for solving semidefinite programs via low-rank factorization. *Mathematical Programming (B)*, 95:329–357, 2003.

[12] R.E. Burkard, M. Dell'Amico, and S. Martello. *Assignment problems*. SIAM, 2009.

[13] E. Chlamtac. *Non-local Analysis of SDP based approximation algorithms*. PhD thesis, Princeton University, USA, 2009.

[14] E. de Klerk. *Aspects of semidefinite programming: interior point algorithms and selected applications*. Kluwer Academic Publishers, 2002.

[15] 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.

[16] E. de Klerk, D.V. Pasechnik, and J.P. Warners. On approximate graph colouring and Max-$k$-Cut algorithms based on the $\vartheta$-function. *Journal of Combinatorial Optimization*, 8(3):267–294, 2004.

[17] R.J. Duffin. Infinite programs. *Ann. Math. Stud.*, 38:157–170, 1956.

[18] I. Dukanovic and F. Rendl. Copositive programming motivated bounds on the stability and chromatic numbers. *Mathematical Programming*, 121:249–268, 2010.

[19] A. Frieze and M. Jerrum. Improved approximation algorithms for Max $k$-Cut and Max Bisection. *Algorithmica*, 18(1):67–81, 1997.

[20] M. Fukuda, M. Kojima, K. Murota, and K. Nakata. Exploiting sparsity in semidefinite programming via matrix completion. I: General framework. *SIAM Journal on Optimization*, 11(3):647–674, 2000.

[21] B. Ghaddar, M. Anjos, and F. Liers. A branch-and-cut algorithm based on semdefinite programming for the minimum $k$-partition problem. *Annals of Operations Research*, 2008.

[22] M. X. Goemans. Semidefinite programming in combinatorial optimization. *Mathematical Programming*, 79:143–162, 1997.

[23] M.X. Goemans and D.P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the ACM*, 42:1115–1145, 1995.

[24] G. Golub and C. F. Van Loan. *Matrix computations. 3rd ed.* Baltimore, MD: The Johns Hopkins Univ. Press. , 1996.

[25] M. Grötschel, L. Lovász, and A. Schrijver. *Geometric algorithms and combinatorial optimization*. Springer, Berlin, 1988.

[26] N. Gvozdenović. *Approximating the stability number and the chromatic number of a graph via semidefinite programming*. PhD thesis, University of Amsterdam, 2008.

[27] N. Gvozdenović and M. Laurent. Computing semidefinite programming lower bounds for the (fractional) chromatic number via block-diagonalization. *SIAM Journal on Optimization*, 19(2):592–615, 2008.

[28] N. Gvozdenović and M. Laurent. The operator $\Psi$ for the chromatic number of a graph. *SIAM Journal on Optimization*, 19(2):572–591, 2008.

[29] C. Helmberg. Semidefinite programming. *European Journal of Operational Research*, 137:461–482, 2002.

[30] C. Helmberg. Numerical validation of SBmethod. *Mathematical Programming*, 95:381–406, 2003.

[31] C. Helmberg, B. Mohar, S. Poljak, and F. Rendl. A spectral approach to bandwidth and separator problems in graphs. *Linear and Multilinear Algebra*, 39:73–90, 1995.

[32] C. Helmberg and F. Rendl. A spectral bundle method for semidefinite programming. *SIAM Journal on Optimization*, 10:673–696, 2000.
[33] J.-B. Hiriart-Urruty and A. Seeger. A variational approach to copositive matrices. *SIAM Review*, forthcoming.
[34] F. Jarre and F. Rendl. An augmented primal-dual method for linear conic problems. *SIAM Journal on Optimization*, 20:808–823, 2008.
[35] D. Karger, R. Motwani, and M. Sudan. Approximate graph colouring by semidefinite programming. *Journal of the ACM*, 45:246–265, 1998.
[36] S.E. Karisch and F. Rendl. Semidefinite Programming and Graph Equipartition. *Fields Institute Communications*, 18:77–95, 1998.
[37] S. Khanna, N. Linial, and S. Safra. On the hardness of approximating the chromatic number. *Combinatorica*, 20:393–415, 2000.
[38] M. Kocvara and M. Stingl. On the solution of large-scale SDP problems by the modified barrier method using iterative solvers. *Mathematical Programming*, 95:413–444, 2007.
[39] M. Laurent and F. Rendl. Semidefinite programming and integer programming. In K. Aardal, G.L. Nemhauser, and R. Weismantel, editors, *Discrete Optimization*, pages 393–514. Elsevier, 2005.
[40] E.L. Lawler, J.K. Lenstra, A.H.G. Rinnooy Kan, and D.B. (eds.) Shmoys. *The traveling salesman problem, a guided tour of combinatorial optimization*. Wiley, Chicester, 1985.
[41] A. Lisser and F. Rendl. Graph partitioning using Linear and Semidefinite Programming. *Mathematical Programming (B)*, 95:91–101, 2002.
[42] L. Lovász. On the Shannon capacity of a graph. *IEEE Trans. Inform. Theory*, 25:1–7, 1979.
[43] L. Lovász. Semidefinite programs and combinatorial optimization. In B. A. Reed and C.L. Sales, editors, *Recent advances in algorithms and combinatorics*, pages 137–194. CMS books in Mathematics, Springer, 2003.
[44] L. Lovász and A. Schrijver. Cones of matrices and set-functions and 0-1 optimization. *SIAM Journal on Optimization*, 1:166–190, 1991.
[45] D. Johnson M. Garey, R. Graham and D.E. Knuth. Complexity results for bandwidth minization. *SIAM Journal on Applied Mathematics*, 34:477–495, 1978.
[46] J. Malick, J. Povh, F. Rendl, and A. Wiegele. Regularization methods for semidefinite programming. *SIAM Journal on Optimization*, 20:336–356, 2009.
[47] 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.
[48] T.S. Motzkin and E.G. Straus. Maxima for graphs and a new proof of a theorem of Turán. *Canadian Journal of Mathematics*, 17:533–540, 1965.
[49] K.G. Murty and S.N. Kabadi. Some NP-complete problems in quadratic and nonlinear programming. *Mathematical Programming*, 39:117–129, 1987.
[50] K. Nakata, K. Fujisawa, M. Fukuda, K. Kojima, and K. Murota. Exploiting sparsity in semidefinite programming via matrix completion. II: Implementation and numerical results. *Mathematical Programming*, 95:303–327, 2003.
[51] Y. Nesterov. Quality of semidefinite relaxation for nonconvex quadratic optimization. Technical report, CORE, 1997.
[52] C.H. Papadimitriou. The NP-completeness of the bandwidth minimization problem. *Computing*, 16:263–270, 1976.
[53] J. Povh. *Applications of semidefinite and copositive programming in combinatorial optimization*. PhD thesis, Universtity of Ljubljana, Slovenia, 2006.
[54] J. Povh and F. Rendl. A copositive programming approach to graph partitioning. *SIAM Journal on Optimization*, 18(1):223–241, 2007.
[55] J. Povh and F. Rendl. Copositive and semidefinite relaxations of the quadratic assignment problem. *Discrete Optimization*, 6(3):231–241, 2009.
[56] J. Povh, F. Rendl, and A. Wiegele. A boundary point method to solve semidefinite programs. *Computing*, 78:277–286, 2006.

[57] F. Rendl. Semidefinite relaxations for integer programming. In M. Jünger Th.M. Liebling D. Naddef G.L. Nemhauser W.R. Pulleyblank G. Reinelt G. Rinaldi and L.A. Wolsey, editors, *50 years of integer programming 1958-2008*, pages 687–726. Springer, 2009.

[58] F. Rendl, G. Rinaldi, and A. Wiegele. Solving max-cut to optimality by intersecting semidefinite and polyhedral relaxations. *Mathematical Programming*, 212:307–335, 2010.

[59] A. Schrijver. A comparison of the Delsarte and Lovász bounds. *IEEE Transactions on Information Theory*, IT-25:425–429, 1979.

[60] A. Schrijver. *Combinatorial optimization. Polyhedra and efficiency. Vol. B*, volume 24 of *Algorithms and Combinatorics*. Springer-Verlag, Berlin, 2003.

[61] H. D. Sherali and W. P. Adams. A hierarchy of relaxations between the continuous and convex hull representations for zero-one programming problems. *SIAM Journal on Discrete Mathematics*, 3(3):411–430, 1990.

[62] H. D. Sherali and W. P. Adams. A hierarchy of relaxations and convex hull characterizations for mixed-integer zero-one programming problems. *Discrete Appl. Math.*, 52(1):83–106, 1994.

[63] J. Sturm. Theory and algorithms of semidefinite programming. In T. Terlaki H. Frenk, K.Roos and S. Zhang, editors, *High performance optimization*, pages 1–194. Springer Series on Applied Optimization, 2000.

[64] J. Sun, S. Boyd, L. Xiao, and P. Diaconis. The fastest mixing Markov process on a graph and a connection to a maximum variance unfolding problem. *SIAM Reviev*, 48(4):681–699, 2006.

[65] M.J. Todd. A study of search directions in primal-dual interior-point methods for semidefinite programming. *Optimization Methods and Software*, 11:1–46, 1999.

[66] K.-C. Toh. Solving large scale semidefinite programs via an iterative solver on the augmented systems. *SIAM Journal on Optimization*, 14:670 – 698, 2003.

[67] L. Vandenberghe and S. Boyd. Semidefinite programming. *SIAM Review*, 38:49–95, 1996.

[68] A. Widgerson. Improving the performance guarantee for approximate graph colouring. *Journal of the ACM*, 30:729– 735, 1983.

[69] A. Wiegele. *Nonlinear optimization techniques applied to combinatorial optimization problems*. PhD thesis, Alpen-Adria-Universität Klagenfurt, Austria, 2006.

[70] H. Wolkowicz, R. Saigal, and L. Vandenberghe (eds.). *Handbook of semidefinite programming*. Kluwer, 2000.

[71] X. Zhao, D. Sun, and K. Toh. A Newton CG augmented Lagrangian method for semidefinite programming. *SIAM Journal on Optimization*, 20:1737–1765, 2010.