In Mathematical Programming, 79 (1997), 143-161.

Semidefinite Programming in Combinatorial Optimization [1]

Michel X. Goemans

MIT, Department of Mathematics, Room 2-392, Cambridge, MA 02139, USA.
E-mail: goemans@math.mit.edu.

We discuss the use of semidefinite programming for combinatorial optimization problems. The main topics covered include (i) the Lovász theta function and its applications to stable sets, perfect graphs, and coding theory, (ii) the automatic generation of strong valid inequalities, (iii) the maximum cut problem and related problems, and (iv) the embedding of finite metric spaces and its relationship to the sparsest cut problem.

1 Introduction

Recently, there has been increasing interest in the use of convex optimization techniques and more specifically semidefinite programming in solving combinatorial optimization problems. This started with the seminal work of Lovász [41] on the so-called theta function, and this led Grötschel, Lovász and Schrijver [22,24] to develop the only known (and non-combinatorial) polynomial-time algorithm to solve the maximum stable set problem for perfect graphs. More recently, the development of efficient interior-point algorithms for semidefinite programming, the results of Lovász and Schrijver [44,45] on stronger formulations using semidefinite programming, improved approximation algorithms for the maximum cut and related problems, and striking hardness of approximation results have spawned much focus on the power (and limitation) of semidefinite programming for combinatorial optimization problems.

In this paper, we give a brief *tour d'horizon* of semidefinite programming in combinatorial optimization. In addition to some of the classical results, we also present a few either very recent or less well known results and observations. In

[1] Part of this work is supported by NSF contract 9623859-CCR, a Sloan Foundation Fellowship, and ARPA Contract N00014-95-1-1246.

particular, we describe the relationship between the Lovász theta function and the Delsarte linear programming approach in Section 3, discuss the use of the dual for solving maximum cut instances in practice in Section 5, and elaborate on the connection between a classical eigenvalue bound and a semidefinite programming approach for the sparsest cut problem in Section 6. Because of space limitations, we can barely scratch the surface, and there are many aspects (e.g. computational) of the area that we will not cover. We refer the reader to Alizadeh [1], Lovász [43] and Rendl [58] for additional coverage of the topic.

2 Preliminaries

In this section, we collect several basic results about (positive semidefinite) matrices and semidefinite programming. Further results will be mentioned as needed. Most of the results on matrices quoted in this paper can be found in standard matrix theory books, such as [36] or [28].

Let M_n denote the cone of $n \times n$ matrices (over the reals), and let S_n denote the subcone of symmetric $n \times n$ matrices. A matrix $A \in S_n$ is said to be *positive semidefinite* if its associated quadratic form $x^T A x$ is nonnegative for all $x \in \mathbb{R}^n$. The positive semidefiniteness of a matrix A will be denoted by $A \succeq 0$; similarly, we write $A \succeq B$ for $A - B \succeq 0$. The cone of positive semidefinite matrices will be denoted by PSD_n. The following statements are equivalent for a symmetric matrix A (see e.g. [36]): (i) A is positive semidefinite, (ii) all eigenvalues of A are nonnegative, and (iii) there exists a matrix B such that $A = B^T B$ (Cholesky decomposition). (iii) gives a representation of $A = [a_{ij}]$ as a *Gram matrix*: there exist vectors v_i such that $a_{ij} = v_i^T v_j$ for all i, j. A symmetric positive semidefinite matrix A can be expressed as LDL^T in $O(n^3)$ elementary operations (where L is lower triangular and D is diagonal), and this leads to a Cholesky decomposition (provided square roots can be computed).

Given $A, B \in M_n$, we consider the (Frobenius) inner product $A \bullet B$ defined by $A \bullet B = Tr(A^T B) = \sum_i \sum_j A_{ij} B_{ij}$. The quadratic form $x^T A x$ can thus also be written as $A \bullet (xx^T)$. Since the extreme rays of PSD_n are of the form xx^T, we derive that $A \bullet B \geq 0$ whenever $A, B \succeq 0$. We can also similarly derive Fejer's theorem which says that PSD_n is self-polar, i.e. $PSD_n^* := \{A \in S_n : A \bullet B \geq 0$ for all $B \succeq 0\} = PSD_n$.

Semidefinite programs are linear programs over the cone of positive semidefinite matrices. They can be expressed in many equivalent forms, e.g.

$$\begin{aligned} SDP \; &= \inf \; C \bullet Y \\ &\text{subject to:} \end{aligned} \tag{1}$$

$$
\begin{aligned}
& A_i \bullet Y = b_i \qquad\qquad\qquad i = 1, \cdots, m \\
& Y \succeq 0.
\end{aligned}
$$

In general a linear program over a pointed closed convex cone K is formulated as $z = \inf\{c^T x : Ax = b, x \in K\}$, and its dual (see [53]) is $w = \sup\{b^T y : A^T y + s = c, s \in K^*\}$ where $K^* := \{a : a^T b \geq 0 \text{ for all } b \in K\}$. Weak duality always holds: $c^T x - y^T b = (A^T y + s)^T x - y^T A x = s^T x \geq 0$ (since $x \in K$ and $s \in K^*$) for any primal feasible x and dual feasible y. If we assume that A has full row rank, $\{x \in \text{int}K : Ax = b\} \neq \emptyset$, and $\{(y, s) : A^T y + s = c, s \in \text{int } K^*\} \neq \emptyset$, then $z = w$ (strong duality) and both the primal and dual problems attain their optimum value. In the case of semidefinite programs, the dual to (1) is $\sup\{\sum_{i=1}^n b_i y_i : \sum_i y_i A_i \preceq C\}$.

Semidefinite programs can be solved (more precisely, approximated) in polynomial time within any specified accuracy either by the ellipsoid algorithm [22,24] or more efficiently through interior-point algorithms. For the latter, we refer the reader to [53,1,65] and to the recent article by Kojima [34] for the latest developments. To be precise, these algorithms are polynomial only for "well-behaved" instances (e.g., if we can give a priori estimates on the sizes of primal and dual solutions that are polynomial in the size of the input, see [1]). The above algorithms produce a strictly feasible solution (or slightly infeasible for some versions of the ellipsoid algorithm) and, in fact, the problem of deciding whether a semidefinite program is feasible (exactly) is still open. However, we should point out that since $\begin{pmatrix} 1 & x \\ x & a \end{pmatrix} \succeq 0$ is equivalent to $|x| \leq \sqrt{a}$, a special case of semidefinite programming feasibility is the square-root sum problem: given positive integers $a_1, \cdots, a_n$ and k, decide whether $\sum_{i=1}^n \sqrt{a_i} \leq k$. The complexity of this problem in the Turing machine model is still open (but the problem is easy in the "unit-cost algebraic RAM", see Malajovich [48] and Tiwari [64]).

Many of the semidefinite programs that arise in combinatorial optimization can also be viewed as eigenvalue bounds [1]. The literature on such bounds is vast, and we refer the reader to a comprehensive survey by Mohar and Poljak [51]. In certain cases, the semidefinite programs can be strengthened by adding valid inequalities. We will see several examples in the forthcoming sections. We would also like to refer the reader to [39] for a discussion on eigenvalue optimization in general.

3 Lovász's Theta Function

Given a graph $G = (V, E)$, a stable (or independent) set is a subset S of vertices such that no two vertices of S are adjacent. The maximum cardinality of a stable set is the stability number (or independence number) of G and is denoted by $\alpha(G)$. In a seminal paper [41], Lovász proposed an upper bound on $\alpha(G)$ known as the theta function $\vartheta(G)$. The theta function can be expressed in many equivalent ways: as an eigenvalue bound, as a semidefinite program, or in terms of orthogonal representations. In this section, we describe some of these formulations, the quality of the resulting approximation, and connections to perfect graphs and coding theory. For simplicity, we restrict our attention to the unweighted case (as defined above), although most results generalize to the weighted case. We refer the reader to the original paper [41], to Chapter 9 in Grötschel et al. [24], or to the survey by Knuth [33] for additional details.

As an eigenvalue bound, $\vartheta(G)$ can be derived as follows. Consider $P = \{A \in S_n : a_{ij} = 1 \text{ if } (i,j) \notin E \text{ (or } i = j)\}$. If there exists a stable set of size k, the corresponding principal submatrix of any $A \in P$ will be J_k, the all ones matrix of size k. By a classical result on interlacing of eigenvalues for symmetric matrices (see [28]), we derive that $\lambda_{max}(A) \geq \lambda_{max}(J_k) = k$ for any $A \in P$, where $\lambda_{max}(\cdot)$ denotes the largest eigenvalue. As a result, $\min_{A \in P} \lambda_{max}(A)$ is an upper bound on $\alpha(G)$, and this is one of the equivalent formulations of Lovász's theta function.

This naturally leads to a semidefinite program. Indeed, the largest eigenvalue of a matrix can easily be formulated as a semidefinite program: $\lambda_{max}(A) = \min\{t : tI - A \succeq 0\}$. This follows from the fact that the eigenvalues of $tI - A$ are precisely $t - \lambda_i$ where $\{\lambda_i\}$ denote the eigenvalues of A. In order to express $\vartheta(G)$ as a semidefinite program, we observe that $A \in P$ is equivalent to $A - J$ being generated by E_{ij} for $(i,j) \in E$, where all entries of E_{ij} are zero except for (i,j) and (j,i). Thus, we can write

$$
\begin{aligned}
\vartheta(G) &= \min t \\
&\quad \text{subject to:} \\
&\qquad tI + \sum_{(i,j)\in E} x_{ij} E_{ij} \succeq J.
\end{aligned}
$$

By strong duality, we can also write:

$$
\begin{aligned}
\vartheta(G) &= \max J \bullet Y && && (2) \\
&\quad \text{subject to:} \\
&\qquad y_{ij} = 0 && (i,j) \in E && (3) \\
&\qquad I \bullet Y = 1 && (i.e. \;\; Tr(Y) = 1) && (4) \\
&\qquad Y \succeq 0. && && (5)
\end{aligned}
$$

Lovász's first definition of $\vartheta(G)$ was in terms of orthonormal representations. An *orthonormal representation* of G is a system $v_1, \cdots, v_n$ of unit vectors in $\mathbb{R}^n$ such that v_i and v_j are orthogonal (i.e. $v_i^T v_j = 0$) whenever i and j are not adjacent. The value of the orthonormal representation is $z = \min_{c:||c||=1} \max_{i \in V} \frac{1}{(c^T u_i)^2}$. This is easily seen to be an upper bound on $\alpha(G)$ (since $||c||^2 \geq \sum_{i \in S} (c^T u_i)^2 \geq |S|/z$ for any stable set S). Taking the minimum value over all orthonormal representations of G, one derives another expression for $\vartheta(G)$ as was shown by Lovász [41]. This result can be restated in a slightly different form. If x denotes the incidence vector of a stable set then we have that

$$\sum_i (c^T v_i)^2 x_i \leq 1. \tag{6}$$

In other words, the *orthonormal representation constraints* (6) are valid inequalities for $STAB(G)$, the convex hull of incidence vectors of stable sets of G. Grötschel et al. [23] show that if we let $TH(G) = \{x : x \text{ satisfies (6) and } x \geq 0\}$, then $\vartheta(G) = \max\{\sum_i x_i : x \in TH(G)\}$. Yet more formulations of ϑ are known (it seems all paths lead to ϑ!); we strongly urge the reader to read Lovász's original article or [23,24] for additional results.

Schrijver [59] proposed a strengthening of $\vartheta(G)$ by adding simple inequalities. We describe this improved upper bound on $\alpha(G)$ in terms of the various formulations discussed above (other formulations of $\vartheta(G)$ can also be similarly improved). The validity of these formulations follow easily from the same arguments as before.

Theorem 1 (Schrijver [59]) *$\alpha(G) \leq \vartheta'(G) \leq \vartheta(G)$ where $\vartheta'(G)$ is equal to*

$$\begin{aligned}
&\min\{\lambda_{max}(A) : a_{ij} \geq 1 \textit{ for } (i,j) \notin E, A = [a_{ij}] \in S_n\} \\
&= \max\{J \bullet Y : y_{ij} = 0 \textit{ for } (i,j) \in E, y_{ij} \geq 0 \textit{ for } (i,j) \notin E, Tr(Y) = 1, Y \succeq 0\} \\
&= \min \max_{i \in V} \{\frac{1}{(c^T u_i)^2} : u_i^T u_j \leq 0 \textit{ for } (i,j) \notin E, ||u_i|| = 1 \textit{ for } i \in V, ||c|| = 1\}.
\end{aligned}$$

3.1 Perfect Graphs

A graph G is called *perfect* if, for every induced subgraph G', its chromatic number is equal to the size of the largest clique in G' (see [21,42] for details). Even though perfect graphs have been the focus of intense study, there are some basic questions which are still open. The strong perfect graph conjecture of Berge is that a graph is perfect if and only if it does not contain an odd cycle of length at least five or its complement. It is not even known if the recognition problem of deciding whether a graph is perfect is in P or is NP-complete. However, the theta function gives some important characterizations (but not a "good" or NP$\cap$co-NP characterization) of perfect graphs.

Theorem 2 (Grötschel et al. [23]) *The following are equivalent:*

- *G is perfect,*
- $TH(G) = \{x \geq 0 : \sum_{i \in C} x_i \leq 1$ *for all cliques* $C\}$
- *$TH(G)$ is polyhedral.*

Moreover, even though recognizing perfect graphs is still open, one can find a largest stable set in a perfect graph in polynomial time by computing the theta function using semidefinite programming (Grötschel et al. [22,24]); similarly one can solve the weighted problem, or find the chromatic number or a largest clique. Observe that if we apply this algorithm to a graph which is not necessarily perfect, we would either find a largest stable set or have a proof that the graph is not perfect.

3.2 Quality of approximations based on ϑ

For perfect graphs, we have seen that $\vartheta(G) = \alpha(G)$. Unfortunately, for general graphs, $\vartheta(G)$ can provide a fairly poor upper bound on $\alpha(G)$, as was established recently. In this section, we discuss the quality of the approximation given by $\vartheta(G)$.

In [41], Lovász showed that for any graph G on n vertices, we have that $\vartheta(G)\vartheta(\bar{G}) \geq n$ (with equality if G is vertex-transitive; see also [59]). Thus, for any G, $\max(\vartheta(G), \vartheta(\bar{G})) \geq \sqrt{n}$, while, for a random graph (each edge being selected with probability 0.5 independently), $\max(\alpha(G), \alpha(\bar{G})) = O(\log n)$ with high probability. In fact, $\vartheta(G) = \Theta(\sqrt{n})$ for random graphs [29]. Until quite recently, this was the largest gap known between α and ϑ. However, Feige [17] has shown the existence of graphs for which $\vartheta(G)/\alpha(G) \geq \Omega(n^{1-\epsilon})$ for any $\epsilon > 0$. His construction uses "randomized graph products". See also Karger et al. [30], Szegedy [63], Alon and Kahale [4] for related results. The fact that $\vartheta(G)$ does not provide a good approximation is not too surprising given the recent result of Håstad [25] showing that the stable set problem is hard to approximate within $n^{1-\epsilon}$ for any $\epsilon > 0$ unless NP=co-R (co-R is the class of languages L for which there exists a polynomial-time randomized algorithm which always accepts elements of L and rejects elements not in L with probability at least 0.5).

Regarding the complementary problem of finding a vertex cover of minimum cardinality, Kleinberg and Goemans [32] have shown that $n - \vartheta(G)$ (where n is the number of vertices) can be arbitrarily close to half the size of the minimum vertex cover (i.e. $n - \alpha(G)$), thus not improving in the worst-case the linear programming bound [27]. Very recently, Lagergren and Russell [35] have also shown that the same holds for Schrijver's $n - \vartheta'(G)$.

3.3 Coding Theory

Lovász's theta function provides interesting results for several coding theory problems. We first discuss the Shannon capacity, and then relate the theta function to Delsarte's linear programming approach [15].

The strong product $G \cdot H$ of $G = (V, E)$ and $H = (W, F)$ is the graph whose vertex set is the cartesian product of V and W and (v, w) is adjacent to (v', w') if v is adjacent or equal to v' and w is adjacent or equal to w'. Given a graph $G = (V, E)$ in which the vertices represent symbols of an alphabet and $(a, b) \in E$ if the symbols a and b cannot be distinguished, the maximum number of distinguishable words that can be written with k symbols is equal to $\alpha(G^k)$, where $G^k = G \cdot G \cdots G$ (k times). The *Shannon capacity* (also called the zero-error capacity) [60] of a graph, denoted by $\Theta(G)$, is equal to $\sup_k \alpha(G^k)^{1/k}$. This quantity appears to be hard to compute even for very small graphs (although it is not known to be NP-hard). Lovász showed that $\vartheta(G)$ provides an upper bound on $\Theta(G)$. This for example implies that $\Theta(G) = \alpha(G)$ for perfect graphs, a result which follows directly from Shannon's early work. For most graphs (but not for all, see the discussion in McEliece [49]), $\vartheta(G)$ provides the best known upper bound on $\Theta(G)$. In particular, for an odd cycle C_n with n vertices, Lovász computed $\vartheta(C_n) = n/(1 + 1/\cos(\pi/n))$, and this implies the celebrated result that $\Theta(C_5) = \sqrt{5}$. However, the exact value for $\Theta(C_7)$ is still unknown.

In order to show that $\vartheta(G) \geq \Theta(G)$, Lovász first proved that $\vartheta(G \cdot H) = \vartheta(G)\vartheta(H)$. This immediately implies that $\alpha(G^k) \leq \vartheta(G^k) = \vartheta(G)^k$, which gives the desired inequality. We should point out that Lovász's proof does not generalize to Schrijver's $\vartheta'(G)$, i.e. $\vartheta'(G)$ is not guaranteed to be an upper bound on $\Theta(G)$.

In certain cases when the graph G has a great deal of symmetry (to be defined formally below), the theta function (as well as Schrijver's ϑ') reduces to a linear programming problem. Such situations arise in coding theory, and more precisely in *association schemes* (see [46,11]). Consider graphs whose adjacency matrix can be written as $\sum_{i \in M} D_i$ where $M \subseteq \{1, \cdots, l\}$ and $D_0, D_1, \cdots, D_l$ are $n \times n$ $0-1$ symmetric matrices such that

(i) $D_0 = I$,
(ii) $\sum_{i=0}^{l} D_i = J$,
(iii) there exist p_{ijk} ($0 \leq i, j, k \leq l$) such that $D_i D_j = D_j D_i = \sum_{k=0}^{n} p_{ijk} D_k$.

For concreteness, we consider one such example. Given $a, b \in \{0,1\}^n$, the Hamming distance $H(a,b)$ between a and b is simply the number of coordinates in which they differ. Let $A(n,d) = \max\{|S| : H(a,b) \geq d$ for all distinct $a, b \in S\}$. $A(n,d)$ represents the maximum number of codewords of a binary code of length n and minimum distance d. Such a (so-called error-correcting) code can correct any number of errors less than $d/2$ introduced during transmission (see MacWilliams and Sloane [46] for background material). $A(n,d)$ can be viewed as the stability number of the graph $G_{n,d}$ with vertex set $\{0,1\}^n$ and two strings being adjacent if their Hamming distance is between 1 and d-1. $G_{n,d}$ arises from the *Hamming association scheme*, in which adjacency in D_i corresponds to pairs of strings having Hamming distance exactly i.

In a seminal paper, Delsarte [15] introduced a beautiful and powerful linear programming approach to upper bound $\alpha(G)$ for graphs arising from association schemes. In the case of the Hamming scheme, given any binary code S of length n with minimum distance d, consider the vector x with 2^n components defined by $x_v = \frac{1}{|S|}|\{(a,b) \in S \times S : a - b = v \pmod 2\}|$ for any $v \in \{0,1\}^n$. x is called the *inner distribution* of S. Observe that $x_0 = 1$, and $\sum_v x_v = |S|$. Moreover, for any $w \in \{0,1\}^n$, we have that

$$\sum_{v \in \{0,1\}^n} (-1)^{v^T w} x_v = \frac{1}{|S|} \sum_{(a,b) \in S \times S} (-1)^{(a-b)^T w} = \frac{1}{|S|} \sum_{(a,b) \in S \times S} (-1)^{a^T w} (-1)^{b^T w}$$
$$= \frac{1}{|S|} \left(\sum_{a \in S} (-1)^{a^T w} \right)^2 \geq 0,$$

by definition of x_v and the fact that $(-1)^k$ only depends on the parity of k. (For any $w \in \{0,1\}^n$, the vector $(-1)^{v^T w}$ is actually an eigenvector of $G_{n,d}$ (for any d).) We can thus obtain an upper bound on $\alpha(G_{n,d})$ by solving

$$\begin{aligned} &\max \sum_v x_v && (7) \\ &\text{subject to:} && \\ &\quad \sum_{v \in \{0,1\}^n} (-1)^{v^T w} x_v \geq 0 && w \in \{0,1\}^n \\ &\quad x_0 = 1, x_v \geq 0 && v \in \{0,1\}^n \\ &\quad x_v = 0 && 0 < |v| < d, \end{aligned}$$

where $|v|$ represents the weight of v (the number of 1's in v). This is a huge linear program in which the constraint matrix has only ± 1 coefficients. A more compact equivalent relaxation can be obtained by observing that, because of symmetry, we can assume that x_v depends only on the weight of v. At the same time, this allows to write one constraint for each possible weight for w. This gives a linear program with $n+1$ variables and $n+1$ constraints, but with fairly nasty coefficients (depending on so-called Krawtchouk polynomials). For an explicit upper bound on $A(n,d)$ based on this linear programming approach, see [50,49]. In general, for association schemes, the linear program obtained can be reduced to $l+1$ variables and constraints, where $l+1$ denotes the number of matrices D_i.

Somewhat surprisingly, Schrijver [59] has shown that the above upper bound is precisely $\vartheta'(G_{n,d})$ and that if we were to relax the nonnegativity constraints on x_v we would obtain precisely Lovász's theta function $\vartheta(G_{n,d})$ (see also McEliece [49]). We will now sketch this equivalence. The fact that the D_j's commute imply that they share a system of eigenvectors (see e.g. [28]), i.e. $D_j = V\Lambda^{(j)}V^T$ where $\Lambda^{(j)}$ is a diagonal matrix with the eigenvalues of D_j, and V is independent of j. This implies that the (so-called Bose-Mesner) algebra spanned by the D_j's consists of all matrices of the form $V\Gamma V^T$ where Γ is a linear combination of the $\Lambda^{(j)}$'s. Furthermore, since the vector u of all 1's is the unique eigenvector of J (which is spanned by the D_j's) corresponding to a nonzero eigenvalue, u must also be an eigenvector of D_j and thus D_j has a constant row sum that we denote by d_j (which is the largest eigenvalue of D_j). Suppose that in formulation (2) of ϑ, we restrict our attention to matrices Y in this Bose-Mesner algebra: $Y = (\sum_{j=0}^{l} x_j D_j)/n$ (where we have normalized by n for simplicity). Observe that $nTr(Y) = \sum_{j=0}^{l} x_j Tr(D_j) = x_0 Tr(I) = nx_0$. Thus (4) reduces to $x_0 = 1$, while (3) reduces to $x_j = 0$ for $j \in M$. Since the eigenvalues of Y are simply $(\sum_{j=0}^{l} x_j \lambda_i^{(j)})/n$ $(i = 1, \cdots, n)$, (5) reduces to $\sum_{j=0}^{l} x_j \lambda_i^{(j)} \geq 0$ for all i. Finally, since $J \bullet D_j = nd_j$, we derive that

$$
\begin{aligned}
\vartheta(G) \ \geq \max & \sum_{j\notin M} d_j x_j && && (8)\\
\text{subject to:} & && && \\
& \sum_{j\notin M} \lambda_i^{(j)} x_j \geq 0 && i = 1, \cdots, n && (9)\\
& x_0 = 1. && &&
\end{aligned}
$$

So far, we have only used the commutativity of the D_j's, and not the fact that $D_i D_j$ is also in the Bose-Mesner algebra (as is stipulated in the definition of association schemes). In fact, if we do not assume this latter condition but only assume commutativity of the D_j's, the value of this linear program may be strictly smaller than $\vartheta(G)$ (contradicting an informal claim of McEliece [49]). For example, for any regular graph G, we can select E_1 and E_2 to be the adjacency matrices of G and its complement (and E_1 and E_2 commute if G is regular), and the above linear program (with only one nontrivial variable) can easily be seen to have value $1 - \lambda_{max}(E_2)/\lambda_{min}(E_2)$ which can be strictly less than $\alpha(G)$ (consider the (perfect) cycle C_6 for example). However, if $\bar{G}$ is edge-transitive (i.e. for every pair of edges of $\bar{G}$, there exists an automorphism mapping one to the other), then one can easily argue that there exists an optimum solution Y to (2) in the Bose-Mesner algebra, and hence we will have equality in (8).

Schrijver proved that we also have equality in (8) if G arises from an association scheme. This is based on additional properties of eigenvalues of association schemes. He similarly showed that, for association schemes, $\vartheta'(G)$ is equal to the value of the linear program (8) in which one adds the constraints $x_j \geq 0$ for $j \notin M$.

Finally, the size of the linear program (8) (or augmented with nonnegative constraints) can be much reduced for association schemes. Indeed, one can show (see e.g. [46]) that there are only $l+1$ different eigenspaces for the D_j's, and thus there are are only $l+1$ distinct constraints (9). (The constraint corresponding to the largest eigenvalue is actually redundant and can be removed; the number of constraints is thus only l.) Furthermore, the resulting LP after adding the inequalities $x_j \geq 0$ for $j \notin M$ takes precisely the same form as Delsarte's LP (7) (once generalized to association schemes)! See [15,59,49] for details.

4 Deriving Valid Inequalities

Lovász and Schrijver [44,45] have proposed a technique for automatically generating stronger and stronger formulations for integer programs. Because of space limitation, we can only briefly describe their approach. We also refer the reader to Sherali and Adams [61], Balas et al. [7], Lovász [43] for additional results and related work.

Let $P = \{x \in \mathbb{R}^n : Ax \geq b, 0 \leq x \leq 1\}$, and let $P_0 = conv(P \cap \{0,1\}^n)$ denote the convex hull of $0-1$ solutions. Suppose we multiply a valid inequality $\sum_i c_i x_i - d \geq 0$ for P by either $1 - x_j \geq 0$ or by $x_j \geq 0$. We obtain a quadratic inequality that we can linearize by replacing $x_i x_j$ by a new variable y_{ij}. But we haven't used yet the fact that we are only interested in 0-1 solutions. Observing that $x_i^2 = x_i$ if $x_i \in \{0,1\}$, we can also replace x_i by y_{ii}, hence obtaining a linear ("matrix") inequality on the entries of Y. Let $M(P)$ denote the set of all symmetric matrices satisfying all the matrix inequalities that can be derived in this way, and let $N(P) = \{x : Y \in M(P), x = diag(Y)\}$, where $diag(Y)$ denotes the diagonal of Y; thus $N(P)$ is a projection of $M(P)$ and conversely $M(P)$ can be viewed as an extended formulation for $N(P)$. By construction, we have that $P_0 \subseteq N(P) \subseteq P$.

Lovász and Schrijver study the operator $N^k(\cdot)$ obtained by repeating $N(\cdot)$ k times, and show that for any $P \subseteq \mathbb{R}^n$ we have $N^n(P) = P_0$. They also prove numerous results on the stable set polytope $STAB(G)$. They introduce the N-index of a valid inequality for $STAB(G)$ as the least k such that this inequality is valid for $N^k(FRAC(G))$, where $FRAC(G) = \{x : x_i + x_j \leq 1 \text{ if } (i,j) \in E, x_i \geq 0 \text{ for all } i \in V\}$. To give a brief sample of their results, they show that the N-index of a clique constraint on k vertices is $k-2$, the N-index of an odd hole constraint is 1, and the N-index of an odd antihole constraint on $2k+1$ vertices is k.

They also consider a much stronger (and not so well understood) operator involving semidefinite constraints. Observe that, for any 0-1 solution x, the matrix Y defined above as xx^T must satisfy $Y - diag(Y)diag(Y)^T = 0$. This is again an (intractable) quadratic inequality but it can be relaxed to $Y - diag(Y)diag(Y)^T \succeq 0$. Viewing $Y - diag(Y)diag(Y)^T$ as a Schur complement (see e.g. [28]), this is equivalent to

$$\begin{bmatrix} 1 & diag(Y)^T \\ diag(Y) & Y \end{bmatrix} \succeq 0. \tag{10}$$

As a result, defining $M_+(P)$ as $\{Y \in M(P) \text{ satisfying } (10)\}$ and $N_+(P) = \{x : Y \in M_+(P), x = diag(Y)\}$, we have that $P_0 \subseteq N_+(P) \subseteq N(P) \subseteq P$ and optimizing a linear objective function over $N_+(P)$ can be done via semidefinite programming. As for $N(\cdot)$, we can define $N_+^k(P)$ and the N_+-index of a valid inequality for P_0. Lovász and Schrijver show that the equivalence between (weak) optimization and (weak) separation [22,24] implies that one can optimize (up to arbitrary precision) in polynomial time over N_+^k for any fixed value of k given a separation oracle for P. For the stable set polytope, they show that all clique, odd hole, odd antihole, odd wheel, and orthonormal representation constraints have N_+-index equal to 1, implying the polynomial-time solvability of the maximum stable set problem in any graph for which these inequalities are sufficient (including perfect graphs, t-perfect graphs, etc.).

Since N_+ is at least as strong as N, we know that $N_+^n(P) = P_0$; however, it is not known if substantially fewer repetitions of N_+ would be sufficient to obtain P_0. To the best of our knowledge, no explicit valid inequality has been proved to have unbounded N_+-index for any problem (even though they should exist, unless P=NP). Consider for example the following (simple looking) polytope considered by Laurent et al. [37]:

$$P = \{x \in \mathbb{R}^n : x_i + x_j + x_k \leq 2 \text{ for all } i,j,k, 0 \leq x \leq 1\}.$$

Correspondingly, $P_0 = \{0 \leq x \leq 1 : \sum_{i=1}^n x_i \leq 2\}$. Using similar arguments as in [45], one can show that $\sum_{i=1}^n x_i \leq 2$ has an N-index no less than $n-3$, but its N_+-index is unknown. Could it be bounded? Or logarithmic in n? A logarithmic bound would follow if one could show that it has bounded N_+-

index if we start from $P = \{x \in \mathbb{R}^n : \sum_{i\in S} x_i \leq 2$ for all $|S| = n/2, 0 \leq x \leq 1\}$.

Another very interesting open problem is related to the matching polytope (the convex hull of incidence vectors of matchings, which can also be viewed as the stable set polytope of the line graph). Consider the Edmonds constraints: $\sum_{i\in S} x_i \leq (|S|-1)/2$ for $|S|$ odd. Their N-index is unbounded (as a function of $|S|$) as was shown by Lovász and Schrijver [45] (and an indirect consequence of a result of Yannakakis [66]). However, their N_+-index is unknown and could possibly be bounded.

5 The Maximum Cut Problem

Given a graph $G = (V, E)$, the cut $\delta(S)$ induced by vertex set S consists of the set of edges with exactly one endpoint in S. In the NP-hard maximum cut problem (MAX CUT), we would like to find a cut of maximum total weight in a weighted undirected graph. The weight of $\delta(S)$ is $w(\delta(S)) = \sum_{e\in\delta(S)} w_e$. Throughout this section, we assume that the weights are nonnegative. For a comprehensive survey of the MAX CUT problem, the reader is referred to Poljak and Tuza [57].

As with the stable set problem, semidefinite programming seems to provide a (semi)definite advantage over linear programming for MAX CUT. A classical linear programming relaxation of the problem (involving cycle constraints, and based on the fact that any cycle intersects a cut in an even number of edges) can be arbitrarily close to twice the optimum value [55]. However, semidefinite programming leads to a much better bound in the worst-case, as was shown by the author and Williamson [20]. In this section, we discuss this approach.

The maximum cut problem can be formulated as an integer quadratic program. If we let $y_i = 1$ if $i \in S$ and $y_i = -1$ otherwise, the value of the cut $\delta(S)$ can be expressed as $\sum_{(i,j)\in E} w_{ij}\frac{1}{2}(1 - y_i y_j)$. Thus, in the spirit of the previous section, suppose we consider the matrix $Y = [y_i y_j]$. This is a positive semidefinite rank one matrix with all diagonal elements equal to 1. Relaxing the rank one condition, we derive a semidefinite program giving an upper bound SDP on OPT:

$$\begin{aligned} SDP \; &= \max \tfrac{1}{2} \sum_{(i,j)\in E} w_{ij}(1 - y_{ij}) && (11) \\ &\text{subject to:} \\ &\qquad y_{ii} = 1 && i \in V \\ &\qquad Y = [y_{ij}] \succeq 0. \end{aligned}$$

It is convenient to write the objective function in terms of the (weighted) *Laplacian* matrix $L(G) = [l_{ij}]$ of G: $l_{ij} = -w_{ij}$ for all $i \neq j$ and $l_{ii} = \sum_j w_{ij}$. For any matrix Y, we have $L(G) \bullet Y = \sum_{(i,j)\in E} w_{ij}(y_{ii} + y_{jj} - 2y_{ij})$ (in particular, if $Y = yy^T$ then we obtain the classical equality $y^T L(G) y = \sum_{(i,j)\in E} w_{ij}(y_i - y_j)^2$). As a result, the objective function can also be expressed as $\frac{1}{4} L(G) \bullet Y$.

The dual of this semidefinite program is $SDP = \frac{1}{4}\min\{\sum_j d_j : Diag(d) \succeq L(G)\}$, where $Diag(d)$ is the diagonal matrix having d as diagonal and all other entries zero. Manipulating this expression [56], this can also be rewritten as

$$SDP = \frac{1}{4} n \min_{u: \sum_i u_i = 0} \lambda_{max}(L + Diag(u)). \tag{12}$$

This eigenvalue bound was proposed and analyzed by Delorme and Poljak [14,13]. In their study, they conjectured that the worst-case ratio OPT/SDP is $32/(25 + 5\sqrt{5}) \sim 0.88445$ and achieved by the 5-cycle. Even though the values of the semidefinite program (11) and the corresponding eigenvalue bound (12) are the same, it appears that (11) provides more information (similar to the fact that a maximum flow provides more information than a minimum cut). By exploiting (11), Goemans and Williamson [20] derived a randomized algorithm that produces a cut whose expected value is at least $0.87856\,SDP$, implying that $OPT/SDP \geq 0.87856$. We describe their *random hyperplane technique* and their elementary analysis below.

Consider any feasible solution Y to (11). Since Y admits a Gram representation (see preliminaries), there exist vectors $v_i \in \mathbb{R}^d$ (for some $d \leq n$) for $i \in V$ such that $y_{ij} = v_i^T v_j$. The fact that $y_{ii} = 1$ implies that the v_i's have unit norm. Let r be a vector uniformly generated from the unit sphere in $\mathbb{R}^d$, and consider the cut induced by the hyperplane $\{x : r^T x = 0\}$ normal to r, i.e. the cut $\delta(S)$ where $S = \{i \in V : r^T v_i \geq 0\}$. The motivation behind the uniform choice for r is that the set of matrices B such that $B^T B = Y$ is closed under orthogonal transformations (or informally rotations). Furthermore, observe that if the matrix Y is of rank one (and thus corresponding to a cut), this random hyperplane technique would recover the cut with probability 1.

By elementary arguments, one can show that the probability that v_i and v_j are separated is precisely θ/π, where $\theta = \arccos(v_i^T v_j)$ is the angle between v_i and v_j. By linearity of expectation, the expected weight of the cut is exactly given by:

$$E[w(\delta(S))] = \sum_{(i,j)\in E} w_{ij} \frac{\arccos(v_i^T v_j)}{\pi}. \tag{13}$$

Comparing this expression term by term to the objective function of (11) and using the fact that $\arccos(x)/\pi \geq \alpha\frac{1}{2}(1-x)$ where $\alpha = 0.87856\cdots$, we derive that $E[w(\delta(S))] \geq \alpha\frac{1}{4}L(G) \bullet Y$. Hence if we apply the random hyperplane technique to a feasible solution Y of value $\geq (1-\epsilon)SDP$ (which can be obtained in polynomial time), we obtain a random cut of expected value greater or equal to $\alpha(1-\epsilon)SDP \geq 0.87856\,SDP \geq 0.87856\,OPT$. Mahajan and Ramesh [47] have shown that this technique can be derandomized, therefore giving a deterministic 0.87856-approximation algorithm for MAX CUT.

The worst-case value for OPT/SDP is thus somewhere between 0.87856 and 0.88446, and even though this gap is small, it would be very interesting to prove Delorme and Poljak's conjecture that the worst-case is given by the 5-cycle. This would, however, require a new technique. Indeed, Karloff [31] has shown that the analysis of the random hyperplane technique is tight, namely there exists a family of graphs for which the expected weight $E[w(\delta(S)]$ of the cut produced is arbitrarily close to $\alpha\,SDP$.

Instead of comparing (13) and (11) term by term, Nesterov [52] recently proposed a different analysis proving that $E[w(\delta(S))] \geq \frac{2}{\pi}(\frac{1}{4}L(G) \bullet Y)$. Even though the resulting bound of $2/\pi = 0.63661\cdots$ is weaker than 0.87856, the analysis only assumes that $L(G) \succeq 0$ and not the stronger requirement that the weights are nonnegative. Furthermore, it has wider applicability than the term-by-term analysis (see [52]). Letting $\arcsin(Y) = [\arcsin(y_{ij})]$, we can write $\frac{\pi}{2}E[w(\delta(S))] = \frac{1}{2}\sum_{(i,j)\in E} w_{ij}\arccos(v_i^T v_j) = \frac{1}{4}\sum_{(i,j)\in E} w_{ij}(\pi - 2\arcsin(v_i^T v_j)) = \frac{1}{4}L(G) \bullet \arcsin(Y)$. Therefore, to derive Nesterov's result, we need to prove that $L(G) \bullet (\arcsin(Y) - Y) \geq 0$. Assuming that $L(G) \succeq 0$ (which holds if the weights are nonnegative), this result follows from a claim that $\arcsin(Y) \succeq Y$.

In order to show that $\arcsin(Y) \succeq Y$, we need the following classical definitions and results. Given two matrices $A, B \in M_n$, the *Hadamard product* (or Schur product) of A and B, denoted by $A \circ B$, is the entry-wise multiplication of A and B, that is the matrix $C = [c_{ij}]$ such that $c_{ij} = a_{ij}b_{ij}$. The Schur product theorem says that if $A, B \succeq 0$ then $A \circ B \succeq 0$. In particular, this implies that $[a_{ij}^k] \succeq 0$ provided that $A = [a_{ij}] \succeq 0$. Furthermore, if $f(z) = c_0 + c_1 z + c_2 z^2 + \cdots$ is an analytic function with nonnegative coefficients and convergence radius $R > 0$ then $[f(a_{ij})] \succeq 0$ provided that $A = [a_{ij}] \succeq 0$ and $|a_{ij}| < R$. We now derive that $\arcsin(Y) \succeq Y$ from $f(x) = \arcsin(x) - x = \frac{1}{2\cdot 3}x^3 + \frac{1\cdot 3}{2\cdot 4\cdot 5}x^5 + \cdots$.

No better approximation algorithm is currently known for MAX CUT. On the negative side though, Håstad [26] has shown that it is NP-hard to approximate MAX CUT within $16/17 + \epsilon = 0.94117\cdots$ for any $\epsilon > 0$. Furthermore, Håstad shows that if we replace the objective function by $\frac{1}{2}\sum_{(i,j)\in E_1} w_{ij}(1 - y_i y_j) + \frac{1}{2}\sum_{(i,j)\in E_2} w_{ij}(1 + y_i y_j)$, then the resulting problem is NP-hard to approximate within $11/12 + \epsilon = 0.91666\cdots$, while the random hyperplane technique still gives the same guarantee of $\alpha \sim 0.87856$.

Several authors have proposed to strengthen (11) by adding *triangle inequalities*, requiring that, for any $i, j, k \in V$, $\pm y_{ij} \pm y_{ik} \pm y_{jk} \geq -1$ whenever we have an even number of minus signs. One of the motivating factor is that these (relatively simple looking, but still hard to analyze) inequalities are sufficient to describe the cut polytope for the 5-cycle (or any planar graph [8]) provided we consider all triplets (i, j, k). If we denote the resulting upper bound by SDP', no better bound than 0.87856 is known for the worst-case ratio OPT/SDP' in general (see Rendl [58] for special cases). The ratio OPT/SDP' is known to be equal to 0.96 for the complete graph K_5, and instances with a slightly worse gap ($\sim .957$) were obtained by Andress and Cheriyan (private communication). However, in light of Håstad's result and the polynomial solvability of semidefinite programs, worse instances should exist (unless P=NP)!

We have implemented the approximation algorithm and have performed limited computational testing. The results will be published after more extensive computational tests. We would like nevertheless to give a preview of some features of the implementation. First we have noticed that the fact that the algorithm is randomized is a plus; by generating several hyperplanes, one typically gets cuts of weight significantly higher than the expected value; theoretically speaking, it is however difficult to get an a priori estimate of the variance of the weight of the cut (which can be zero even if the dimension is not one, as for an odd cycle). Furthermore, the vectors $\{v_i\}$ typically lie in a very low-dimensional space (see [9,54,2] and [20] for theoretical explanations), and as a result, one can often enumerate all possible hyperplane cuts. Finally, and more importantly, the dual semidefinite program can be exploited very nicely in a branch-and-bound scheme, and this often allows to prove optimality of the cut produced for instances with up to 100 vertices. The dual can be reinterpreted as follows:

$$SDP = \frac{1}{4} \min \left\{ \left|\left| \sum_i u_i \right|\right|^2 : u_i^T u_j = w_{ij} \text{ for all } (i,j) \in E \right\},$$

and the vectors u_i can be obtained by a Cholesky decomposition. The fact that this expression is an upper bound on OPT is obvious: given a cut $\delta(S)$, its weight is equal to $\sum_{i\in S}\sum_{j\notin S} w_{ij} = (\sum_{i\in S} u_i)^T(\sum_{j\notin S} u_j) = \frac{1}{4}(b+x)^T(b-x) = \frac{1}{4}(||b||^2 - ||x||^2) \leq \frac{1}{4}||b||^2$, where $b = \sum_{i\in V} u_i$ and $x = \sum_{i\in S} u_i - \sum_{j\notin S} u_j$. Observe that we used a trivial bound on $||x||^2 = \sum_k x_k^2 \geq 0$. If we can prove that, no matter how V is partitioned into S and $V-S$, $|x_k|$ is at least δ_k, then we can refine our upper bound to $SDP - \sum_k \delta_k^2$. In particular, if coordinate k of u_i is zero for all but one vector then we can trivially let $\delta_k = |x_k|$. This may seem trivial, but it leads to a powerful branch-and-bound scheme. First, since the u_i's are obtained by a Cholesky decomposition, we can assume that only the first i components of u_i are nonzero. In our branch-and-bound procedure, any node at level $n-l$ corresponds to an assignment $\tau : \{l+1, l+2, \cdots, n\} \rightarrow$

$\{0,1\}$ where $\tau(v)$ indicates if v is constrained to be in S or in $V-S$. Therefore, for cuts corresponding to this node, we can let $\delta_k = |z_k|$ for $k > l$ and $\delta_k = 0$ otherwise, where $z = \sum_{i>l:\tau(i)=1} u_i - \sum_{i>l:\tau(i)=0} u_i$. The main advantage of this procedure is that the computation of these improved upper bounds is negligible (once we have solved the semidefinite program at the root node); we can therefore explore a very large number of nodes (easily a billion). This branch-and-bound code often allows to prove optimality of the best hyperplane cut generated, especially for problems with up to 100 vertices; in these cases, the branch-and-bound procedure typically takes less time than the solution of the initial semidefinite program. Details and extensive computational tests will be given in a forthcoming experimental paper.

The results described in this section have been extended and generalized to other combinatorial optimization problems: the maximum dicut problem and the maximum 2-satisfiability problem [20,18], the problem of coloring 3-colorable graphs [30], the maximum k-cut and maximum bisection problems [19], and the betweenness problem [12].

6 Embeddings of Finite Metric Spaces

We would like to conclude with some open problems related to the power of semidefinite programming for the sparsest cut problem. This is a fascinating area but, unfortunately, we will be able to explore only the tip of the iceberg.

We first collect some results on finite metric spaces; references for most results mentioned here can be found in the new book of Deza and Laurent [16]. A (finite) (semi)-metric $d : V \times V \to \mathbb{R}$ on V satisfies $d(i,i) = 0$ for all $i \in V$, $d_{ij} = d_{ji}$ for all $i,j \in V$, and $d_{ij} + d_{jk} \geq d_{ik}$ for all $i,j,k \in V$. d is said to be l_p-embeddable if there exists $x_i \in \mathbb{R}^k$ $(i \in V)$ for some k such that $d(i,j) = ||x_i - x_j||_p$; similarly, d is l_p-embeddable with distortion c if there exists $x_i \in \mathbb{R}^d$ such that $d(i,j) \leq ||x_i - x_j||_p \leq c\, d(i,j)$. Any (finite) metric d is l_∞-embeddable, and deciding if d is l_1-embeddable is NP-hard. Any l_2-embeddable metric is in fact l_1-embeddable (a random projection onto a line similar to the random hyperplane technique can be seen to imply this result). It is also known that d is l_2-embeddable iff $P = [p_{ij}] = [d_{1i}^2 + d_{1j}^2 - d_{ij}^2] \succeq 0$ (this is independent of the choice of $1 \in V$), and thus the l_2-embeddability can be tested in polynomial time. Moreover, finding the (square of the) smallest distortion for an l_2-embedding of d is therefore a semidefinite program [40]: $\min\{t : d_{ij}^2 \leq x_{ij} \leq t d_{ij}^2$ for all $i,j, [x_{1i} + x_{1j} - x_{ij}] \succeq 0\}$. Bourgain [10] has shown that any finite metric on n points can be embedded with distortion $O(\log n)$ into l_2 (and thus also l_1); it would be nice to prove this result from semidefinite programming duality. An interesting open problem in this area is whether any l_1-embeddable metric on n points can be embedded into l_2 with distortion $O(\sqrt{\log n})$ (and this would be tight because of the d-dimensional hypercube).

Let $K = \{X = [x_{ij}] \in S_n^0 : [x_{1i} + x_{1j} - x_{ij}] \succeq 0\}$, where S_n^0 denotes symmetric $n \times n$ matrices with zero elements on the diagonal. Thus d is l_2-embeddable if $[d_{ij}^2] \in K$. If $D = [d_{ij}] \in K$ and d is a metric then d is called a *negative type metric*; the requirement that d is a metric now translates into the fact that the angle between any three points of the l_2-embedding of the metric $\sqrt{d}$ is either acute or right. Any l_1-embeddable metric can be seen to be of negative type. Finding the best negative type metric subject to linear constraints can therefore be solved in polynomial time through semidefinite programming. (As an exercise, the reader can reinterpret the bounds on the maximum cut problem in terms of negative type metrics.) One can show that K is a pointed closed convex cone, and its polar can be expressed nicely as:

$$K^* = \{Y \in S_n^0 : L(Y) \succeq 0\} \tag{14}$$

where $L(Y)$ is the Laplacian of Y.

In the sparsest cut problem, we would like to find a cut minimizing $\frac{w(\delta(S))}{|S||V-S|}$ in a (nonnegatively weighted) undirected graph $G = (V, E)$. Since the cone generated by incidence vectors of cuts (also called cut metrics) is precisely the l_1-embeddable metrics, the problem reduces to

$$\min_{d \in l_1}\{\sum_{(i,j)\in E} w_{ij} d_{ij} : \sum_i \sum_j d_{ij} = 1\}.$$

If we relax the requirement that d is l_1-embeddable to simply being a metric, we obtain a linear programming relaxation (whose dual is a multicommodity flow problem) of the sparsest cut problem. Leighton and Rao [38] show that this relaxation is always within $O(\log n)$ of the optimum sparsest cut, and recently Linial et al. [40] (see also [6]) used Bourgain's result [10] to generalize this result. Moreover, the logarithmic ratio is tight as is shown by constant degree expander graphs.

If instead we relax the l_1-embeddability of d to membership in K (and not even impose that d is a metric), we obtain the following linear program over the cone K:

$$z = \min\{\sum_{(i,j)\in E} w_{ij} d_{ij} : \sum_i \sum_j d_{ij} = 1, d \in K\}.$$

By duality over cones (see preliminaries) and (14), this is equivalent to

$$z = \max\{\lambda : L(G) - \lambda L(J - I) \succeq 0\}.$$

Using the facts that $L(J-I) = nI - J$ and that the vector of all 1's is an eigenvector of both $L(G)$ and J, this dual can be shown to be equivalent to a well-known eigenvalue bound [3,5,62]: $z = \frac{1}{n}\lambda_2(L(G))$. See these references for relations between z and the value of the sparsest cut.

Finally, we could impose that d is a negative type metric, therefore getting a lower bound on the sparsest cut which is stronger than both the LP relaxation and the eigenvalue bound. Using duality and (14), we can express this bound in many different ways, and we leave this as an exercise for the reader. The most interesting question though is the worst-case ratio between the sparsest cut and this lower bound. If one could show that negative type metrics can be embedded into l_2 with $O(\sqrt{\log n})$ distortion (or possibly even into l_1 within a constant), this would give a worst-case ratio that is $O(\sqrt{\log n})$ (resp. constant). This is a very intriguing question.

Acknowledgement

I am grateful to Levent Tuncel, David Williamson, and the referee for pointing out several inaccuracies in a previous draft. Thanks also to Dan Spielman for stimulating discussions on coding theory and the sparsest cut problem.

References

[1] F. Alizadeh. Interior point methods in semidefinite programming with applications to combinatorial optimization. *SIAM Journal on Optimization*, 5:13–51, 1995.

[2] F. Alizadeh, J.-P. Haeberly, and M. Overton. Complementarity and nondegeneracy in semidefinite programming. Technical report, Courant Institute of Mathematical Sciences, 1995. To appear in *Mathematical Programming*.

[3] N. Alon. Eigenvalues and expanders. *Combinatorica*, 6:83–96, 1986.

[4] N. Alon and N. Kahale. Approximating the independence number via the θ function. Manuscript, 1994.

[5] N. Alon and V. Milman. λ_1, isoperimetric inequalities for graphs and superconcentrators. *Journal of Combinatorial Theory B*, 38:73–88, 1985.

[6] Y. Aumann and Y. Rabani. An $O(\log k)$ approximate min-cut max-flow theorem and approximation algorithm. *SIAM Journal on Computing*, To appear.

[7] 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.

[8] F. Barahona and A. Mahjoub. On the cut polytope. *Mathematical Programming*, 36:157–173, 1986.

[9] A. I. Barvinok. Problems of distance geometry and convex properties of quadratic maps. *Discrete and Computational Geometry*, 13:189–202, 1995.

[10] J. Bourgain. On Lipschitz embedding of finite metric spaces in Hilbert space. *Israel Journal of Mathematics*, 52:46–52, 1985.

[11] A. Brouwer and W. Haemers. Association schemes. In R. Graham and M. G. amd L. Lovász, editors, *Handbook of Combinatorics*, pages 747–771. Elsevier Science, 1995.

[12] B. Chor and M. Sudan. A geometric approach to betweenness. In *Proc. of 3rd Europ. Symp. on Algs.*, volume LNCS 979 of *Lecture Notes in Computer Science*, pages 227–237, 1995.

[13] C. Delorme and S. Poljak. Combinatorial properties and the complexity of a max-cut approximation. *European Journal of Combinatorics*, 14:313–333, 1993.

[14] C. Delorme and S. Poljak. Laplacian eigenvalues and the maximum cut problem. *Mathematical Programming*, 62:557–574, 1993.

[15] P. Delsarte. An algebraic approach to the association schemes of coding theory. *Philips Research Reports, Supplement*, 10, 1973.

[16] M. Deza and M. Laurent. *Geometry of cuts and metrics.* Algorithms and Combinatorics. Springer-Verlag, 1997.

[17] U. Feige. Randomized graph products, chromatic numbers, and the Lovász θ-function. In *Proc. of the 27th ACM Symp. on Theory Comput.*, pages 635–640, 1995.

[18] U. Feige and M. X. Goemans. Approximating the value of two prover proof systems, with applications to MAX 2SAT and MAX DICUT. In *Proc. 3rd Israel Symp. Theory Comput. and Syst.*, pages 182–189, 1995.

[19] A. Frieze and M. Jerrum. Improved approximation algorithms for MAX k-CUT and MAX BISECTION. *Algorithmica*, 18:67–81, 1997.

[20] M. X. Goemans and D. P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *J. ACM*, 42:1115–1145, 1995. A preliminary version entitled ".878-Approximation Algorithms for MAXCUT and MAX2SAT" has appeared in Proc. 26th ACM Symp. on Theory of Comput., 422–431, 1994.

[21] M. Golumbic. *Algorithmic Graph Theory and Perfect Graphs.* Academic Press, 1980.

[22] M. Grötschel, L. Lovász, and A. Schrijver. The ellipsoid method and its consequences in combinatorial optimization. *Combinatorica*, 1:169–197, 1981.

[23] M. Grötschel, L. Lovász, and A. Schrijver. Relaxations of vertex packing. *Journal of Combinatorial Theory B*, 40:330–343, 1986.

[24] M. Grötschel, L. Lovász, and A. Schrijver. *Geometric Algorithms and Combinatorial Optimization.* Springer-Verlag, Berlin, 1988.

[25] J. Håstad. Clique is hard to approximate within $n^{1-\epsilon}$. In *Proc. 37th Symp. on Found. of Comp. Sci.*, pages 627–636, 1996.

[26] J. Håstad. Some optimal inapproximability results. In *Proc. of the 29th ACM Symp. on Theory Comput.*, 1997.

[27] D. S. Hochbaum. Approximation algorithms for the set covering and vertex cover problems. *SIAM Journal on Computing*, 11:555–556, 1982.

[28] R. Horn and C. Johnson. *Matrix Analysis.* Cambridge University Press, 1985.

[29] F. Juhász. The asymptotic behavior of Lovász' ϑ function for random graphs. *Combinatorica*, pages 153–155, 1982.

[30] D. Karger, R. Motwani, and M. Sudan. Approximate graph coloring by semidefinite programming. In *Proc. 35th Symp. on Found. of Comp. Sci.*, pages 2–13, 1994.

[31] H. Karloff. How good is the Goemans-Williamson MAX CUT algorithm. In *Proc. of the 28th ACM Symp. on Theory Comput.*, pages 427–434, 1996.

[32] J. Kleinberg and M. X. Goemans. The Lovász theta function and a semidefinite programming relaxation of vertex cover. *SIAM Journal on Discrete Mathematics.* To appear.

[33] D. Knuth. The sandwich theorem. *Electronic Journal of Combinatorics*, 1, 1994. http://ejc.math.gatech.edu:8000/Journal/journalhome.html.

[34] M. Kojima. Interior-point methods for semidefinite programming. This volume, 1997.

[35] J. Lagergren and A. Russell. Vertex cover eludes the Schrijver function. Manuscript, 1997.

[36] P. Lancaster and M. Tismenetsky. *The Theory of Matrices.* Academic Press, Orlando, FL, 1985.

[37] M. Laurent, S. Poljak, and F. Rendl. Connections between the semidefinite relaxations of the max-cut and stable set problems. *Mathematical Programming*, 1996. To appear.

[38] F. Leighton and S. Rao. An approximate max-flow min-cut theorem for uniform multicommodity flow problems with applications to approximation algorithms. In *Proc. 29th Symp. on Found. of Comp. Sci.*, pages 422–431, 1988.

[39] A. Lewis and M. Overton. Eigenvalue optimization. *Acta Numerica*, 5:149–190, 1996.

[40] N. Linial, E. London, and Y. Rabinovich. The geometry of graphs and some of its algorithmic applications. *Combinatorica*, 15:215–246, 1995.

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

[42] L. Lovász. Perfect graphs. In L. Beineke and R. Wilson, editors, *Selected Topics in Graph Theory*, volume 2, pages 55–87. Academic Press, 1983.

[43] L. Lovász. Combinatorial optimization: Some problems and trends. Technical Report 92-53, DIMACS, 1992.

[44] L. Lovász and A. Schrijver. Matrix cones, projection representations, and stable set polyhedra. In *Polyhedral Combinatorics*, volume 1 of *DIMACS series in Disc. Math. and Theor. Comp. Sci.*, pages 1–17. AMS, 1989.

[45] L. Lovász and A. Schrijver. Cones of matrices and setfunctions, and 0-1 optimization. *SIAM Journal on Optimization*, 1:166–190, 1991.

[46] F. MacWilliams and N. Sloane. *The theory of error correcting codes*. North-Holland, 1977.

[47] S. Mahajan and H. Ramesh. Derandomizing semidefinite programming based approximation algorithms. In *Proc. 36th Symp. on Found. of Comp. Sci.*, pages 162–169, 1995.

[48] G. Malajovich. An effective version of Kronecker's theorem on simultaneous Diophantine approximation. Technical report, City University of Hong Kong, 1996.

[49] R. McEliece. The bounds of Delsarte and Lovász, and their applications to coding theory. In G. Longo, editor, *Algebraic Coding Theory and Applications*, pages 107–178. Springer-Verlag, 1979.

[50] R. McEliece, E. Rodemich, H. Rumsey, and L. Welch. New upper bounds on the rate of a code via the Delsarte-McWilliams inequalities. *IEEE Transactions on Information Theory*, IT-23:157, 1977.

[51] B. Mohar and S. Poljak. Eigenvalue methods in combinatorial optimization. In R. Brualdi, S. Friedland, and V. Klee, editors, *Combinatorial and Graph-Theoretic Problems in Linear Algebra*, volume 50 of *The IMA Volumes in Mathematics and its Applications*, pages 107–151. Springer-Verlag, 1993.

[52] Y. Nesterov. Quality of semidefinite relaxation for nonconvex quadratic optimization. Manuscript, 1997.

[53] Y. Nesterov and A. Nemirovskii. *Interior Point Polynomial Methods in Convex Programming*. SIAM, Philadelphia, PA, 1994.

[54] G. Pataki. On cone-LP's and semi-definite programs: Facial structure, basic solutions, and the simplex method. GSIA Working Paper WP 1995-03, Carnegie-Mellon University, 1995.

[55] S. Poljak. Polyhedral and eigenvalue approximations of the max-cut problem. In G. Halász et al., editor, *Sets, Graphs, and Numbers*, volume 60 of *Colloquia mathematica Societatis Janos Bolyai*, pages 569–581. North-Holland, 1992.

[56] S. Poljak and F. Rendl. Nonpolyhedral relaxations of graph-bisection problems. *SIAM Journal on Optimization*, 5:467–487, 1995.

[57] S. Poljak and Z. Tuza. Maximum cuts and largest bipartite subgraphs. In W. Cook, L. Lovász, and P. Seymour, editors, *Combinatorial Optimization*, DIMACS series in Disc. Math. and Theor. Comp. Sci. AMS, 1995.

[58] F. Rendl. Semidefinite programming and combinatorial optimization. Notes for a lecture series given at the University of Pisa, 1997.

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

[60] C. Shannon. The zero error capacity of a channel. *IRE Transactions on Information Theory*, 2:8–19, 1956.

[61] 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.

[62] A. Sinclair and M. Jerrum. Approximate counting, uniform generation and rapidly mixing markov chains. *Information and Computation*, 82:93–133, 1989.

[63] M. Szegedy. A note on the theta number of Lovász and the generalized Delsarte bound. In *Proc. 35th Symp. on Found. of Comp. Sci.*, pages 36–39, 1994.

[64] P. Tiwari. A problem that is easier to solve on the unit-cost algebraic RAM. *Journal of Complexity*, 8:393–397, 1992.

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

[66] M. Yannakakis. Expressing combinatorial optimization problems by linear programs. In *Proc. 29th Symp. on Found. of Comp. Sci.*, pages 223–228, 1988.