A LINEAR PROGRAMMING APPROACH TO SEMIDEFINITE PROGRAMMING PROBLEMS*

KARTIK KRISHNAN† AND JOHN E. MITCHELL‡

Abstract. Until recently, the study of interior point methods has dominated algorithmic research in semidefinite programming (SDP). From a theoretical point of view, these interior point methods offer everything one can hope for; they apply to all SDP's, exploit second order information and offer polynomial time complexity. Still for practical applications with many constraints k, the number of arithmetic operations, per iteration is often too high. This motivates the search for other approaches, that are suitable for large k and exploit problem structure.

Recently Helmberg and Rendl developed a scheme that casts SDP's with a constant trace on the primal feasible set as eigenvalue optimization problems. These are convex nonsmooth programming problems and can be solved by bundle methods. In this paper we propose a linear programming framework to solving SDP's with this structure. Although SDP's are *semi infinite* linear programs, we show that only a small number of constraints, namely those in the bundle maintained by the spectral bundle approach, bounded by the square root of the number of constraints in the SDP, and others polynomial in the problem size are typically required. The resulting LP's can be solved rather quickly and provide reasonably accurate solutions. We present numerical examples demonstrating the efficiency of the approach on combinatorial examples.

Key words. semidefinite programming, linear programming, spectral bundle, eigenvalue optimization, combinatorial optimization, cutting plane approach

AMS subject classifications. Primary, 65F15, 65K05; Secondary, 52A41, 90C05, 90C06, 90C22, 90C27, 90C34, 90C57

1. Introduction. Semidefinite programming (SDP) has been one of the most exciting and active research areas in optimization recently. This tremendous activity was spurred by the discovery of important applications in combinatorial optimization, control theory, the development of efficient interior point algorithms for solving SDP problems, and the depth and elegance of the underlying optimization theory. Excellent survey articles for SDP include Vandenberghe and Boyd [42], the SDP handbook edited by Wolkowicz et al [44], Helmberg [15] and Todd [40].

Since the seminal work of Alizadeh [1] and Nesterov and Nemirovskii [34], the study of interior point methods has dominated algorithmic research in semidefinite programming. However, for practical applications with many constraints k, the number of arithmetic operations per iteration is often too high. The main computational task here, is the factorization of the *Schur complement* matrix M of size k, in computing the search direction. Typically this is a dense matrix, and a Cholesky factorization would require $\frac{k^3}{3}$ arithmetic operations. Moreover this matrix is to be recomputed in each iteration and this takes $O(kn^3 + k^2n^2)$ arithmetic operations, which is the most expensive operation in each iteration where n is the problem size. For most problems, the constraint matrices have a special structure, which can be exploited to speed up the computation of this matrix. In particular in combinatorial applications, these constraints often have a rank one structure. This reduces the computation time of M to $O(kn^2 + k^2n)$ operations. Benson, Ye and Zhang [3] have proposed a dual scaling algorithm that exploits this rank one feature, and the sparsity in the dual slack

*This work was supported in part by NSF grant numbers CCR–9901822 and DMS-9872019

†Department of Mathematical Sciences, Rensselaer Polytechnic Institute, 110 8th Street, Troy, New York, 12180 (kartis@rpi.edu).

‡Department of Mathematical Sciences, Rensselaer Polytechnic Institute, 110 8th Street, Troy, New York, 12180 (mitchj@rpi.edu).

matrix. However even in their approach the matrix M is dense, and the necessity to store and factorize this matrix limits the applicability of these methods to problems with about 3000 constraints on a well equipped work station.

Consider the semidefinite programming problem

$$\begin{array}{llcll} \min & C \bullet X & & & \\ \text{subject to} & \mathcal{A}(X) & = & b & \qquad (SDP) \\ & X & \succeq & 0, & \end{array}$$

with dual

$$\begin{array}{lllcll} \max & b^T y & & & & \\ \text{subject to} & \mathcal{A}^T y & + & S & = & C \qquad (SDD) \\ & & & S & \succeq & 0 \end{array}$$

where $X, S \in \mathcal{S}^n$, the space of real symmetric $n \times n$ matrices. We define

$$C \bullet X \quad = \quad \text{Trace}(C^T X) \quad = \quad \textstyle\sum_{i,j=1}^n C_{ij} X_{ij}$$

where $\mathcal{A} : \mathcal{S}^n \to I\!R^k$ and $\mathcal{A}^T : I\!R^k \to \mathcal{S}^n$ are of the form

$$\mathcal{A}(X) \quad = \quad \begin{bmatrix} A_1 \bullet X \\ \vdots \\ A_k \bullet X \end{bmatrix} \quad \text{and} \quad \mathcal{A}^T y \quad = \quad \textstyle\sum_{i=1}^k y_i A_i$$

with $A_i \in \mathcal{S}^n, i = 1, \ldots, k$. We assume that $A_1, \ldots, A_k$ are linearly independent in $\mathcal{S}^n$. $C \in \mathcal{S}^n$ is the cost matrix, $b \in I\!R^k$ the RHS vector. The matrix X is constrained to positive semidefinite (*psd*) expressed as $X \succeq 0$. This is equivalent to requiring that $d^T X d \geq 0$, $\forall d$. On the other hand $X \succ 0$ denotes a positive definite (*pd*) matrix, i.e. $d^T X d > 0$, $\forall d \neq 0$. $\mathcal{S}_+^n$ and $\mathcal{S}_{++}^n$ denote the space of symmetric *psd* and *pd* matrices respectively. Also $\text{diag}(X)$ is a vector whose components are the diagonal elements of X, and $\text{Diag}(d)$ is a diagonal matrix, with the components of d. In the succeeding sections we use $\text{Trace}(X)$ and $\text{tr}(X)$ interchangeably, to denote the trace of the symmetric matrix X. $\lambda_{min}(M)$ denotes the minimum eigenvalue of the matrix $M \in \mathcal{S}^n$. An excellent reference for these linear algebra preliminaries is Horn and Johnson [21].

Recently Helmberg and Rendl [16] have developed a scheme known as the *spectral bundle*, a nonsmooth optimization technique applicable to eigenvalue optimization problems. This method is suitable for large k and exploits problem structure. We provide a short overview of this scheme in section 3. However it is only a first order method with no polynomial bound on the number of arithmetic operations. Other large scale methods include Burer et al [5, 6, 7], who formulate SDP's as nonconvex programming problems using low rank factorizations of the primal matrix X. The authors exploit the bound on the rank of optimal matrix X in ensuring that $X = VV^T$, where V is a $n \times r$ matrix for some small r. Vanderbei et al [43] factor the primal matrix X as $L\text{Diag}(d)L^T$, where L is unit triangular and $d \in I\!R^n$. The constraint that $X \succeq 0$ is replaced with the requirement that $d \geq 0$. The authors show that d is a concave function and give computational results for this reformulation. Finally we must mention that Burer et al [8, 9] have come up with attractive heuristics for max cut and maximum stable set problems, where they solve (SDP) with an additional restriction on the rank of the primal matrix X.

The spectral bundle method requires the following assumption 1.1, which enables recasting (SDD) as an eigenvalue optimization problem. Since we need the bundle method in the LP approach to generate some of our linear constraints, we shall make the following assumption too.

ASSUMPTION 1.

$$\mathcal{A}(X) \quad = \quad b \quad implies \quad trX \quad = \quad a \tag{1.1}$$

for some constant $a \geq 0$.

We shall also make the following strict feasibility assumption.

ASSUMPTION 2. *Both* (SDP) *and* (SDD) *have strictly feasible points, namely the sets* $\{X \in \mathcal{S}^n : \mathcal{A}(X) = b, X \succ 0\}$ *and* $\{(y, S) \in I\!R^k \times \mathcal{S}^n : \mathcal{A}^T y + S = C, S \succ 0\}$ *are nonempty.*

This assumption guarantees that both (SDP) and (SDD) attain their optimal solutions X^* and (y^*, S^*), and their optimal values are equal, i.e. $C \bullet X^* = b^T y^*$. Thus the duality gap $X^* S^* = 0$ at optimality.

A large class of semidefinite programs, in particular several important relaxations of combinatorial optimization problems, can be formulated to satisfy (1.1), such as max cut, Lovasz theta, semidefinite relaxations of box constrained quadratic programs etc.

Since most of the combinatorial problems we discuss in this paper are graph problems, we define some terminology. A graph is a pair $G = (V, E)$, where V and E are the vertices and edges of the graph respectively. We denote the edge between vertices i and j by $\{i, j\}$. Henceforth n and m refer to the number of vertices and edges of the graph G. Also $j \in \delta(i)$ refers to all the vertices j adjacent to vertex i. $L \in \mathcal{S}^n$ refers to the *Laplacian* matrix of the graph. Let the edges have a weight vector $w = (w_{ij}) \in I\!R^E$ associated with them. We shall assume that all our graphs are complete, by setting $w_{ij} = 0$ for all non edges $\{i, j\}$. The Laplacian matrix is given by $L = \text{Diag}(Ae) - A$, where A is the weighted adjacency matrix with $A_{ii} = 0$, $\forall i$ and $A_{ij} = w_{ij}$, $\forall \{i, j\} \in E$. Thus the Laplacian matrix L is

$e = [1 \cdots 1]^T$

$$\begin{array}{rcll} L_{ii} & = & \sum_j w_{ij} & \forall i \\ L_{ij} & = & -w_{ij} & i \neq j \end{array}$$

The set of symmetric matrices $\mathcal{S}^n$ is isomorphic to $I\!R^{\frac{n(n+1)}{2}}$. Hence (SDP) is essentially a linear program in $I\!R^{\frac{n(n+1)}{2}}$ variables, except for the convex constraint $X \succeq 0$. Note that the convex constraint $X \succeq 0$ in (SDP) is equivalent to

$$d^T X d \quad = \quad dd^T \bullet X \quad \geq \quad 0 \quad \forall d \in I\!R^n \tag{1.2}$$

These constraints are linear inequalities in the matrix variable X, but there is an *infinite* number of them. Thus SDP is a *semi-infinite* linear programming problem in $I\!R^{\frac{n(n+1)}{2}}$. The term semi-infinite programming derives from the fact that the LP has finitely many variables, with an infinite number of constraints. The survey paper by Hettich and Kortanek [20] discusses theory, algorithms, and applications of semi-infinite programming.

The main objective of this paper is to demonstrate that we can take a finite number of constraints in (1.2), polynomial in the problem size n and yet obtain a reasonable polyhedral approximation of the semidefinite cone.

Since $d^T X d \geq 0$ can be rewritten as $\text{tr}(dd^T X) \geq 0$, the definition of positive semidefiniteness immediately gives the following :

COROLLARY 1.1. *The symmetric $n \times n$ matrix S is positive semidefinite if and only if $S \bullet M \geq 0$ for all symmetric rank one matrices M.*

The paper is organized as follows. Section 2 explains the linear programming approach, section 3 presents a short overview of the bundle approach, section 4 discusses how we can in practice get a lower bound from the LP relaxation, in section 5 we generate the linear inequalities to be used in the LP framework, section 6 presents a cutting plane LP framework to solving SDP, section 7 presents computational results on combinatorial optimization problems which satisfy (1.1), and we conclude with some observations and acknowledgements in sections 8 and 9 respectively.

2. A linear programming formulation. It follows from Corollary 1.1 that (SDD) is equivalent to the following semi-infinite linear programming problem:

$$\begin{array}{llcll} \max & b^T y & & & \\ \text{subject to} & dd^T \bullet (C - \mathcal{A}^T y) & \geq & 0 & \text{for all vectors } d. \end{array} \qquad (LDD)$$

We propose looking at linear programming relaxations of (LDD). Given a finite set of vectors $\{d_i, i = 1, \ldots, m\}$, we obtain the relaxation

$$\begin{array}{llcll} \max & b^T y & & & \\ \text{subject to} & d_i d_i^T \bullet \mathcal{A}^T y & \leq & d_i d_i^T \bullet C & \text{for } i = 1, \ldots, m. \end{array} \qquad (LDR)$$

$$\mathcal{A} = \begin{bmatrix} A_1 \\ A_2 \\ \vdots \\ A_k \end{bmatrix}$$

Since we have k variables, we need at least k constraints for (LDR) to have an extreme point. Since (SDP) has $\frac{n(n+1)}{2}$ variables, a typical finite linear programming relaxation would need at least $O(n^2)$ inequalities to get a basic feasible solution. Since we have far fewer variables in the dual formulation, we typically get smaller linear programs. Another reason for working with (SDD) is that the spectral bundle method, to be introduced in section 3, employs (SDD), recasting it as an eigenvalue optimization problem (3.1).

need
SDP $O(n^2)$ ineq
LDR $\geq k$ ineq

We now derive the linear programming dual to (LDR). We have

$$\begin{aligned} d_i d_i^T \bullet \mathcal{A}^T y &= d_i d_i^T \bullet (\sum_{j=1}^{k} y_j A_j) \\ &= \sum_{j=1}^{k} y_j d_i^T A_j d_i. \end{aligned}$$

Thus, the constraints of (LDR) can be written as

$$\sum_{j=1}^{k} y_j d_i^T A_j d_i \leq d_i^T C d_i \quad \text{for } i = 1, \ldots, m.$$

$\to x_i \sum_{j=1}^{k} y_j d_i^T A_j d_i \leq x_i d_i^T C d_i$

$x_i \geq 0$

It follows that the dual problem is

$$\begin{array}{llcll} \min & \sum_{i=1}^m d_i^T C d_i x_i & & & \\ \text{subject to} & \sum_{i=1}^m d_i^T A_j d_i x_i & = & b_j & \text{for } j = 1, \ldots, k \\ & x & \geq & 0. & \end{array}$$

$\sum_{i=1}^{m} x_i \sum_{j=1}^{k} y_j d_i^T A_j d_i \leq \sum_{i=1}^{m} x_i d_i^T C d_i$

$\sum_{j=1}^{k} y_j \underbrace{\sum_{i=1}^{m} x_i d_i^T A_j d_i}_{b_j} \leq \sum_{i=1}^{m} x_i d_i^T C d_i$

This can be rewritten as

$$\begin{array}{llcl} \min & C \bullet (\sum_{i=1}^m x_i d_i d_i^T) & & \\ \text{subject to} & \mathcal{A}(\sum_{i=1}^m x_i d_i d_i^T) & = & b \\ & x & \geq & 0. \end{array} \qquad (LPR)$$

Linear Programming Relaxation

max min.

LEMMA 2.1. *Any feasible solution x to (LPR) will give a feasible solution X to (SDP).*

Proof. Set $X = \sum_{i=1}^{m} x_i d_i d_i^T$. From (LPR) it is clear that this X satisfies $\mathcal{A}X = b$. Moreover X is psd. To see this

$$\begin{array}{rclcl} d^T X d & = & d^T(\sum_{i=1}^{m} x_i d_i d_i^T)d & = & \sum_{i=1}^{m} x_i (d_i^T d)^2 \\ & \geq & 0 & & \forall d \end{array}$$

where the last inequality follows from the fact that $x \geq 0$. □

The optimal value to (LDR) gives an upper bound on the optimal value of (SDP).

Given an iterate $\bar{y}$ feasible in (LDR), a good way of generating cutting planes $\sum_{j=1}^{k} y_j d^T A_j d \leq d^T C d$ requires searching for d satisfying $dd^T \bullet (C - \mathcal{A}^T \bar{y}) = d^T C d - \sum_{j=1}^{k} \bar{y}_j d^T A_j d \cancel{<} > 0$. One such choice for d would be to look at the eigenvectors d corresponding to the negative eigenvalues of $C - \mathcal{A}^T \bar{y}$, since these satisfy $d^T(C - \mathcal{A}^T \bar{y})d = \lambda(C - \mathcal{A}^T \bar{y}) < 0$, when the d's are normalized.

In Krishnan and Mitchell [27] we develop a cutting plane framework, where we add a number of d corresponding to the most negative eigenvalues of $C - \mathcal{A}^T \bar{y}$ to our finite collection of linear inequalities and solve the new LP relaxation using an interior point approach, with the initial LP's being solved rather cheaply. Continuing in this way, one can produce a sequence of LP's whose solutions converge to the optimal solution of the SDP. We present more details in section 6.

$\min_X \langle C, X \rangle$

subject to $\mathcal{A}(X) = b$

$Tr(X) = a$

$X \succeq 0$

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

$= b^T y - \langle \mathcal{A}^T y, X \rangle + \langle C, X \rangle - \langle S, X \rangle + \lambda(\langle I, X \rangle - a) = b^T y - a\lambda + \langle \lambda I + C - S - \mathcal{A}^T y, X \rangle$

3. The spectral bundle method. Consider the eigenvalue optimization problem (3.1).

$$\max_y \quad a\lambda_{min}(C - \mathcal{A}^T \mathrm{y}) \quad + \quad b^T \mathrm{y} \tag{3.1}$$

Problems of this form are equivalent to the dual of semidefinite programs (SDP), whose primal feasible set has a constant trace, i.e. $\mathrm{Trace}(X) = a$ for all $X \in \{X \succeq 0 : \mathcal{A}(X) = b\}$. This can be easily verified as follows. From the variational characterization of the minimum eigenvalue function, we have $\lambda_{min}(C - \mathcal{A}^T y) = \min_{X:\mathrm{tr}X=1, X \succeq 0} (C - \mathcal{A}^T y) \bullet X$. Thus (3.1) is equivalent to taking the Lagrangian dual of (SDP) with y being the vector of dual variables corresponding to $\mathcal{A}(X) = b$, and observing that $a\lambda_{min}(C - \mathcal{A}^T \mathrm{y}) = \min_{X:\mathrm{tr}X=a, X \succeq 0} (C - \mathcal{A}^T y) \bullet X$.

$\Rightarrow \max_y b^T y - a\lambda$

subject to $\lambda I + C - \mathcal{A}^T y = S \succeq 0$

$\Rightarrow \lambda = \lambda_{max}(\mathcal{A}^T y - C)$.

$-\lambda = \lambda_{min}(C - \mathcal{A}^T y)$

$= (C - \mathcal{A}^T y) \bullet dd^T$

where $(C - \mathcal{A}^T y)d = \lambda_{min} d$

$d^T d = 1$

An excellent survey on eigenvalue optimization appears in Lewis and Overton [29]. The minimum eigenvalue function $\lambda_{min}(.)$ is a nonsmooth concave function. A general scheme to minimize such a function is the bundle method; see Lemarechal [28], Kiwiel [25], Schramm et al [39] and the books by Urruty and Lemarechal [22, 23]. The spectral bundle method specializes the proximal bundle method to eigenvalue optimization problems, and employs a semidefinite nonpolyhedral cutting plane model; it is particularly well suited for large scale problems because of its aggregation possibilities.

The method is due to Helmberg and Rendl [16]. Good references include Helmberg et al [16, 15, 17], Helmberg and Oustry [18]. The following lemma, from Helmberg [15], indicates the equivalence between (SDP) and the eigenvalue optimization problem (3.1).

Lemma 3.1.11 in the book

LEMMA 3.1. *If $\mathcal{A}$ satisfies $\mathcal{A}^T \bar{y} = I$, for some $\bar{y} \in R^k$ then (SDD) is equivalent to (3.1) for $a = \max\{0, b^T \bar{y}\}$. Furthermore, if (SDP) is feasible then all its feasible solutions X satisfy $trX = a$, the primal optimum is attained and is equal to the infimum of (SDD)*

At each step the function value and a subgradient of the function are computed at a specific point y. A cutting plane model of the function is formed using the collected subgradients. For the eigenvalue optimization problem (3.1), the subgradient information is provided by the eigenvectors corresponding to the minimum eigenvalue of $C - \mathcal{A}^T y$. Since the cutting plane model is built from local information from a few previous iterates, the model function is augmented with a regularization term, which imposes a penalty for the displacement from the current iterate. The minimizer of the cutting plane model, obtained by solving a semidefinite program with a concave quadratic cost function, yields the new point. Here the minimum eigenvalue is approximated by means of vectors in the subspace spanned by the bundle P. Without the regularization term, this amounts to solving the following problem (3.2) at each iteration in lieu of (3.1) :

$$\max_y \quad a\lambda_{min}(P^T(C - \mathcal{A}^T y)P) + b^T y \tag{3.2}$$

This is equivalent to solving the following (SDP)

$$\min_{X:\mathrm{tr}(X)=a,\mathcal{A}(PXP^T)=b,X\succeq 0} \quad C \bullet (PXP^T) \tag{3.3}$$

(3.3) implies that we are approximately solving (SDP), by considering only a subset of the feasible X matrices. By keeping the number of columns r in P small the resulting SDP can be solved quickly. Helmberg and Rendl [16] using results from Pataki [36] were able to show that r is bounded by $\sqrt{k}$, i.e. the dimension of the subspace P is roughly bounded by the square root of the number of constraints. Moreover they introduce the concept of an aggregate subgradient, whereby the spectral bundle scheme converges even for restricted bundle sizes. The idea here is to keep all the important cutting planes in the bundle, and aggregate the least important ones. In the extreme case the bundle P consists of one new eigenvector for the minimal eigenvalue alone. The extremal eigenvalues and eigenvectors of $C - \mathcal{A}^T y$ are computed using some iterative scheme such as the *Lanczos* method which involves only matrix vector operations. However (3.2) requires that the dual slack matrix S be positive semidefinite only with respect to a subspace of vectors in bundle P; thus it may be interpreted as a relaxation of (SDD). The optimal solution of this relaxed SDP typically produces an indefinite dual slack matrix. The negative eigenvalues and the corresponding eigenvectors of the slack matrix are used to update the subspace in order to improve the relaxation, and the process is iterated. Every iteration either corresponds to a *serious step* where we update y, or a *null step* where we maintain the old y, but update the model with valuable subgradient information.

In [16], $\min_y a\lambda_{max}(C-\mathcal{A}^T(y))+\langle b,y\rangle$ maximal eigenvalue.

To summarize, the essential idea of the spectral bundle method is the following Todd [40]; it can be regarded as providing an approximation by considering only a subset of the feasible X matrices, using this to improve the dual solution y, and using this in turn to improve the subset of feasible solutions in the primal.

The restricted bundle size ensures that the spectral bundle method is only a *first order method*. Helmberg and Kiwiel [17] were also able to extend the method to problems with bounds. A second order bundle method which converges globally and which enjoys asymptotically a quadratic rate of convergence was recently developed by Oustry [35].

4. Getting a lower bound from our LP relaxation. When (SDP) is a minimization problem, our LP approach gives an upper bound on the SDP objective value. In this section we discuss how we can generate a lower bound from our LP relaxation. The SDP value lies somewhere in between these two LP objective values. The approach makes use of Lemma 3.1.

We want a feasible solution to

$$\begin{array}{lrcll} \max & b^T y & & & \\ \text{s.t.} & S & = & C - \mathcal{A}^T y & \qquad (SDD) \\ & S & \succeq & 0 & \end{array}$$

Our linear programming approach gives a candidate S and y, satisfying the linear constraint but with S not necessarily psd. A procedure to make S psd, generating $b^T y - \bar{\lambda} b^T \bar{y}$, which serves as the lower bound is as follows

$L = b^T y + \langle \Lambda, I - \mathcal{A}^T y \rangle, \Lambda \in \mathbb{R}^{n \times n}$

$= b^T y + \langle \Lambda, I \rangle - \langle \Lambda, \mathcal{A}^T y \rangle$

$= (b - \mathcal{A}(\Lambda))^T y + \langle \Lambda, I \rangle$

$\max_{\Lambda} \operatorname{tr}(\Lambda)$

s.t. $b = \mathcal{A}(\Lambda)$

How to fix $S \succeq 0$

1. Solve the LP

$$\begin{array}{lrcl} \min & b^T y & & \\ \text{s.t.} & \mathcal{A}^T y & = & I \end{array}$$

   Let $\bar{y}$ be the optimal solution.
2. Find the most negative eigenvalue of S. Let $\bar{\lambda}$ denote the absolute value of this eigenvalue.
3. Change y to $y - \bar{\lambda}\bar{y}$, which changes S to $S + \bar{\lambda} I$, a psd matrix. The objective function value is now $b^T y - \bar{\lambda} b^T \bar{y}$, a lower bound on the optimal value of (SDP).

5. A set of linear constraints. A set of linear constraints for (LDR) can be derived from the bundle information used by the spectral bundle method.

At iteration i of the spectral bundle method, we work with the following approximation $\phi_i(y)$ of the convex function $f(y) = a\lambda_{min}(C - \mathcal{A}^T y) + b^T y$

$$\phi_i(y) = \min_{W \in \bar{\mathcal{W}}_i} (C - \mathcal{A}^T y) \bullet W \tag{5.1}$$

where $\bar{\mathcal{W}}_i$ is

$$\bar{\mathcal{W}}_i = \{P_i V P_i^T + \alpha \bar{W}_i \ : \ \operatorname{tr} V + \alpha = a, V \in \mathcal{S}_+^{r_i}, \alpha \geq 0\}.$$

We refer to P_i as the *bundle*, the number of columns r_i of P_i as the *size* of the bundle and to $\bar{W}_i$ as the *aggregate*.

The following result due to Pataki [36] gives a bound on the rank of optimal X matrices.

Theorem 5.1. *There exists an optimal solution X^* with rank r satisfying the inequality $\frac{r(r+1)}{2} \leq k$. Here k is the number of constraints in (SDP).*

Note that if we were not using the aggregate matrix $\bar{W}$, then $P_i V P_i^T$ would be our current approximation of the primal matrix X. To preserve the set of optimal solutions, r_i from Theorem 5.1 should be at least $\max\{r \geq 0 : \frac{r(r+1)}{2} \leq k\}$. Thus P_i is a $n \times r_i$ matrix, with r_i typically bounded by $\sqrt{k}$. We denote the columns of P_i as p_j, $j = 1, \ldots, r_i$. These columns represent a good subspace to approximate the minimum eigenvalue of $C - \mathcal{A}^T y_i$. $\bar{W}_i$ serves as the aggregate and contains the less important subgradient information. This helps to keep the number of columns in P_i small, even lower than $\sqrt{k}$ if necessary, so that the SDP (5.1) can be solved quickly.

When the stopping criteria is met $W^+ := PV^+P^T + \alpha^+\bar{W}$ serves as the approximation to the primal matrix X, where V^+ and α^+ are the final values of V and α.

We propose using the columns of P as the vectors $\{d_j\}$, $j = 1, \ldots, r$ in (LDR). Since the number of vectors r in the bundle P, is $O(\sqrt{k})$, and we need at least k constraints to guarantee a basic feasible solution with k variables in (LDR), we need to look for other constraints as well. These other constraints, labeled as *box* constraints, are problem specific as we shall see in section 5. They are polynomial in the problem size n.

The drawback of using the columns of P as vectors d in (LDR), is these d are dense, leading to a dense linear programming problem. We try to compensate for these dense constraints, by choosing d for our box constraints that are sparse.

The rationale for using the columns of P as d is discussed in detail in section 5.1. We illustrate the LP procedure on the max cut problem (section 5.2), min bisection (section 5.3), **k** equipartition problem (section 5.4), Lovasz theta (section 5.5), and box constrained QP's (section 5.6).

5.1. Rationale for using the columns of P as d. We propose the columns of the bundle P as the vectors $\{d\}$ in (LDR). Note that the columns of P are generated as cutting planes, while solving (3.1). This section gives a stronger motivation for why they may be good cutting planes in the limit. We first describe how the matrix P is updated in each iteration. We will denote the value of P and y in the kth iteration by P^k and y^k respectively.

1. Initially P^0 contains only the eigenvector corresponding to the $\lambda_{min}(\mathcal{A}^T y^0 - C)$.
2. In iteration k, if P^k still contains less than maximum number of columns r allowed, then we add the eigenvector corresponding to $\lambda_{min}(\mathcal{A}^T y^k - C)$, and orthonormalize it, with respect to the vectors already in the bundle. Else we proceed to step 3.
3. If P^k already uses the maximum number of columns r, then method of updating P^k to get P^{k+1} is as follows
   (a) Solve the restricted eigenvalue problem with the quadratic regularization term to obtain α^{k+1} and V^{k+1}.
   (b) Compute a spectral decomposition of V^k namely $V^k = Q \Lambda Q^T$. Then split the eigenvectors of Q into two parts $Q = [Q_1 | Q_2]$, with Q_1 containing the eigenvectors associated with the $r-1$ large eigenvalues of V^k. Let v^{k+1} be the eigenvector corresponding to $\lambda_{min}(C - \mathcal{A}^t y^{k+1})$. P^{k+1} is then computed as

$$P^{k+1} \quad = \quad \mathrm{orth}([P^k Q_1, v^{k+1}])$$

Now the rationale for using the columns of P as the vectors $\{d\}$ is as follows

1. Based on the procedure described above in updating P, we can look at the columns of P as *almost* eigenvectors corresponding to the minimal eigenvalues of $(C - \mathcal{A}^t y)$. We use the word *almost* since in the procedure described above, the vectors in P are eigenvectors corresponding to minimal eigenvalues of $S = (C - \mathcal{A}^T y)$, where the iterates y change only slightly (owing to the quadratic regularization term) in each iteration. As a result the dual slack matrix S changes in each iteration, and orthonormalizing the new vector which is an eigenvector corresponding to the current S w.r.t. those already in the bundle, changes the original eigenvectors (hopefully only slightly).
2. Since X and $S \in \mathcal{S}^n$ and satisfy $XS = 0$ at optimality, they commute and hence are simultaneously diagonalizable. Thus they share the same set of

eigenvectors $\hat{P}$ and have spectral decompositions $\hat{P}\Lambda_X\hat{P}^T$ and $\hat{P}\Lambda_S\hat{P}^T$ respectively. Here $\hat{P}$ is a $n \times n$ orthonormal matrix. Let us rewrite $\hat{P} = [P|\tilde{P}]$, where P is a $n \times r$ orthonormal matrix, essentially our optimal bundle. From Theorem 5.1 we know that only the first r eigenvalues of X are nonzero, with their corresponding r eigenvectors in P. Thus the columns of $\tilde{P}$ form a basis for the null space of X, while those in P are in the null space of the dual slack matrix S at optimality. Thus the columns of the bundle P should be especially useful in ensuring that S is psd. Thus we can write $X = \begin{bmatrix} P & \tilde{P} \end{bmatrix} \begin{bmatrix} V & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} P^T \\ \tilde{P}^T \end{bmatrix} = PVP^T$, which is precisely what the spectral bundle method gives us at optimality, if we disregard the aggregate matrix.

3. At termination the dual slack matrix $S = C - \mathcal{A}^t y$ will be positive semidefinite on the subspace spanned by the columns of P, but may well not be positive semidefinite over the whole of $I\!\!R^n$. However the spectral bundle method works very well in practice, and so it seems to identify an important subspace namely the subspace spanned by the columns of P on which S should be positive semidefinite. Directions d not in the subspace do not seem as important.

We offer empirical evidence for this choice of d in section 7.

5.2. The Max Cut problem. The semidefinite programming relaxation of the max cut problem was proposed by Goemans and Williamson [12] and is

$$\begin{array}{llcl} \max & \frac{L}{4} \bullet X & & \\ \text{subject to} & \operatorname{diag}(X) & = & e \\ & X & \succeq & 0, \end{array} \tag{5.2}$$

with dual

$$\begin{array}{lllcl} \min & e^T y & & & \\ \text{subject to} & -\operatorname{Diag}(y) & + \; S & = & -\frac{L}{4} \\ & & S & \succeq & 0 \end{array} \tag{5.3}$$

Here L is the Laplacian matrix of the graph defined in section 1. Note that the a in $\operatorname{tr}X = a$ is trivially n, the number of nodes in the graph.

Since S is psd, we have $d^T S d = d^T(\operatorname{Diag}(y) - \frac{L}{4})d \geq 0$, $\forall d$. In particular we propose to use the following d for the max cut problem.

MC1 Setting $d = e_i$, $i = 1 \ldots n$, where e_i is the ith standard basis vector for R^n. In particular e_i has a one in the ith position and zeros elsewhere. This generates the constraint $y_i \geq \frac{L_{ii}}{4}$, $i = 1 \ldots n$.

MC2 Setting $d = (e_i + e_j)$ and $(e_i - e_j)$, $\forall \{i,j\} \in E$, gives rise to the constraints $y_i + y_j \geq \frac{Lii}{4} + \frac{Ljj}{4} + \frac{L_{ij}}{2}$ and $y_i + y_j \geq \frac{Lii}{4} + \frac{Ljj}{4} - \frac{Lij}{2}$ respectively. Together these give $y_i + y_j \geq \frac{Lii}{4} + \frac{L_{jj}}{4} + |\frac{Lij}{2}|$.

MC3 The constraints in the bundle namely the columns p_i , $i = 1, \ldots, r$ of the matrix P.

We consider two LP relaxations, $LP1$ containing MC1 and MC3 and $LP2$ containing MC2 in addition to the constraints in $LP1$. Both these relaxations are discussed below.

$LP1$ is

$$
\begin{array}{lrcll}
 & \min & e^T y & & \\
(5.4) & \text{subject to} & y_i & \geq & \frac{L_{ii}}{4} \quad \forall i = 1, \ldots, n \\
 & & \sum_{i=1}^n p_{ji}^2 y_i & \geq & p_j^T \frac{L}{4} p_j \quad \forall j = 1, \ldots, r
\end{array}
$$

with dual

$$
\begin{array}{lll}
 & \max & \sum_{i=1}^n \frac{L_{ii}}{4} x_i + \sum_{j=1}^r w_j \frac{p_j^T L p_j}{4} \\
(5.5) & \text{subject to} & \begin{bmatrix} 1 & & & \uparrow & & \uparrow \\ & \ddots & & p_1^2 & \cdots & p_r^2 \\ & & 1 & \downarrow & & \downarrow \end{bmatrix} \begin{bmatrix} x \\ w \end{bmatrix} = e \\
 & & \begin{bmatrix} x \\ w \end{bmatrix} \geq 0
\end{array}
$$

Here p_{ji} refers to the jth component of vector p_i and p_i^2, $i = 1, \ldots, r$ are vectors obtained by squaring all the components of p_i , $i = 1, \ldots, r$. (5.4) has $n+r$ constraints in all. Note that $x \in I\!R^n$ and $w \in I\!R^r$ are the dual variables corresponding to $y \geq \text{diag}\frac{L}{4}$ and the bundle constraints respectively. To get a solution X to SDP, set $X = \text{Diag}(x) + \sum_{j=1}^r w_j p_j p_j^T$. This matrix X is positive semidefinite since $x \geq 0$ and $w \geq 0$. Moreover

$$
\frac{L}{4} \bullet X = \frac{L}{4} \bullet (\text{Diag}(x) + \sum_{j=1}^r w_j p_j p_j^T) = \sum_{i=1}^n \frac{L_{ii}}{4} x_i + \sum_{j=1}^r w_j \frac{p_j^T L p_j}{4}
$$

This is precisely the objective value in (5.5). We have thus generated the X which could be used in the Goemans and Williamson rounding procedure [12] to generate an approximate solution to the max cut problem.

$LP2$ is

$$
\begin{array}{lrcll}
 & \min & e^T y & & \\
 & \text{subject to} & y_i & \geq & \frac{L_{ii}}{4} \quad \forall i = 1, \ldots, n \\
(5.6) & & y_i + y_j & \geq & \frac{Lii}{4} + \frac{L_{jj}}{4} + |\frac{Lij}{2}| \quad \forall \{i,j\} \in E \\
 & & \sum_{i=1}^n p_{ji}^2 y_i & \geq & p_j^T \frac{L}{4} p_j \quad \forall j = 1, \ldots, r
\end{array}
$$

$LP2$ (5.6) has $m+n+r$ constraints in all. To generate a solution X to SDP, set $X = \text{Diag}(x) + \sum_{j=1}^m w_j d_j d_j^T + \sum_{j=1}^r w_{m+j} p_j p_j^T$. Here d_i , $i = 1, \ldots, m$, corresponding to the edge constraints, are either $d = e_i + e_j$ or $e_i - e_j$ depending on the sign of L_{ij}. It is easy to verify that this X is psd and achieves the dual objective value.

Using the Goemans and Williamson [12] (GW) rounding procedure on the X generated by solving the relaxation $LP2$, we can generate a cut that is at least 0.878 times the $LP2$ objective value. We cannot guarantee that the objective value of relaxation $LP2$ is an upper bound on the maximum cut value. However, in practice the $LP2$ objective is within 1% of the spectral bundle objective value, which incidentally is an upper bound on the optimal SDP value. Thus we have some performance guarantee on the cut produced by solving the LP relaxation $LP2$ followed by the GW randomized rounding procedure.

5.3. The Min Bisection problem. The semidefinite programming relaxation for the min bisection problem was proposed by Frieze and Jerrum [11] and Ye [45] and is

$$\begin{array}{lrcl} \min & \frac{L}{4} \bullet X & & \\ \text{subject to} & \mathrm{diag}(X) & = & e \\ & ee^T \bullet X & = & 0 \\ & X & \succeq & 0, \end{array} \tag{5.7}$$

with dual

$$\begin{array}{lcccccccc} \min & e^T y & & & & & & \\ \text{subject to} & -y_0(ee^T) & - & \mathrm{Diag}(y) & + & S & = & \frac{L}{4} \\ & & & & & S & \succeq & 0 \end{array} \tag{5.8}$$

Note that Frieze and Jerrum [11] had the equipartition constraint as $ee^T \bullet X \leq 0$. But since the optimal X is psd, we must have $e^T X e = ee^T \bullet X \geq 0$ at optimality, which is equivalent to $ee^T \bullet X = 0$.

Here L refers to the Laplacian matrix of the graph. y_0 is the dual variable corresponding to the constraint $ee^T \bullet X = 0$. To get the signs right, we need to take the negative of the objective value of (SDD) to get the optimal solution to the min bisection problem. Again $a = n$.

Since y_0 does not appear in the dual objective function, we rewrite the equipartition constraint in (5.7) as $(ee^T + I) \bullet X = n$. This is true since $\mathrm{Trace} X = I \bullet X = n$.

This gives the SDP

$$\begin{array}{lrcl} \min & \frac{L}{4} \bullet X & & \\ \text{subject to} & \mathrm{diag}(X) & = & e \\ & (ee^T + I) \bullet X & = & n \\ & X & \succeq & 0, \end{array} \tag{5.9}$$

with dual

$$\begin{array}{lcccccccc} \min & ny_0 & + & e^T y & & & & \\ \text{subject to} & S & = & \mathrm{Diag}(y) & + & y_0(ee^T + I) & + & \frac{L}{4} \\ & & & & S & & \succeq & 0 \end{array} \tag{5.10}$$

Since $S = y_0(ee^T + I) + \mathrm{Diag}(y) + \frac{L}{4}$ is p.s.d. we require $d^T S d = d^T(y_0(ee^T + I) + \mathrm{Diag}(y) + \frac{L}{4})d \geq 0$, $\forall d$.

In particular we propose to use the following d for the min bisection problem.

MB1 Setting $d = e_i$, $\forall i = 1 \ldots n$, gives $2y_0 + y_i \geq -\frac{Lii}{4}$, $\forall i = 1 \ldots n$.

MB2 Setting $d = e_i - e_j$, $\forall \{i,j\} \in E$, we obtain $y_i + y_j + 2y_0 \geq \frac{Lij}{2} - \frac{Lii}{4} - \frac{Ljj}{4}$, $\forall \{i,j\} \in E$.

MB3 Setting $d = e_i + e_j$, $\forall \{i,j\} \in E$, we obtain $6y_0 + y_i + y_j \geq -\frac{L_{ij}}{2} - \frac{L_{ii}}{4} - \frac{L_{jj}}{4}$, $\forall \{i,j\} \in E$.

MB4 Setting $d = e$, where e is the all ones vector gives $(n^2 + n)y_0 + \sum_{i=1}^n y_i \geq 0$, since $Le = 0$.

MB5 The constraints in the bundle namely the columns p_i , $i = 1, \ldots, r$ of the matrix P.

The constraints in MB2 provide a lower bound on the LP objective value. To see this sum up all these constraints along a cycle say $\{1, 2, \ldots, n, 1\}$. This gives $ny_0 + \sum_{i=1}^{n} y_i \geq B \bullet \frac{L}{4}$, where B is a matrix with negative ones along the diagonal and ones in certain off diagonal positions.

The resulting LP is

$$
\begin{array}{llrcll}
 & \min & ny_0 + e^T y & & & \\
 & \text{subject to} & 2y_0 + y_i & \geq & -\frac{Lii}{4} & \forall i = 1 \ldots n \\
(5.11) & & 2y_0 + y_i + y_j & \geq & \frac{L_{ij}}{2} - \frac{Lii}{4} - \frac{Ljj}{4} & \forall \{i,j\} \in E \\
 & & 6y_0 + y_i + y_j & \geq & -\frac{L_{ij}}{2} - \frac{L_{ii}}{4} - \frac{L_{jj}}{4} & \forall \{i,j\} \in E \\
 & & (n^2+n)y_0 + \sum_{i=1}^{n} y_i & \geq & 0 & \\
 & & ((p_j^T e)^2 + 1)y_0 + \sum_{i=1}^{n} p_{ji}^2 y_i & \geq & -p_j^T \frac{L}{4} p_j & \forall j = 1, \ldots, r
\end{array}
$$

Note that the optimal face is unbounded, since there is a ray in the direction $y_0 = 1, y_i = -1, i = 1, \ldots, n$. This is not surprising since the dual SDP (5.8) has an unbounded optimal face as well. To observe this set $y_0 \to \infty$ in (5.8). Doing so keeps S psd, and since y_0 does not appear in the objective function, this value remains unchanged. Therefore we impose an upper bound for y_0 say u. This gives the LP

$$
\begin{array}{llrcll}
 & \min & ny_0 + e^T y & & & \\
 & \text{subject to} & 2y_0 + y_i & \geq & -\frac{L_{ii}}{4} & \forall i = 1 \ldots n \\
 & & 2y_0 + y_i + y_j & \geq & \frac{L_{ij}}{2} - \frac{L_{ii}}{4} - \frac{L_{jj}}{4} & \forall \{i,j\} \in E \\
(5.12) & & 6y_0 + y_i + y_j & \geq & -\frac{L_{ij}}{2} - \frac{L_{ii}}{4} - \frac{L_{jj}}{4} & \forall \{i,j\} \in E \\
 & & (n^2+n)y_0 + \sum_{i=1}^{n} y_i & \geq & 0 & \\
 & & ((p_j^T e)^2 + 1)y_0 + \sum_{i=1}^{n} p_{ji}^2 y_i & \geq & -p_j^T \frac{L}{4} p_j & \forall j = 1, \ldots, r \\
 & & y_0 & \leq & u &
\end{array}
$$

Here $p_{ji}, \forall j = 1, \ldots, n$ refers to the jth component of vector $p_i, \forall i = 1, \ldots, r$. (5.12) has $2m + n + r + 1$ constraints in all.

If we set the upper bound constraint, we are in essence solving the following pair of SDP's

$$
\begin{array}{llrcl}
 & \min & \begin{bmatrix} \frac{L}{4} & 0 \\ 0 & u \end{bmatrix} & \bullet & \begin{bmatrix} X & 0 \\ 0 & x_s \end{bmatrix} \\
(5.13) & \text{subject to} & \operatorname{diag}(X) & = & e \\
 & & (ee^T + I) \bullet X + x_s & = & n \\
 & & \begin{bmatrix} X & 0 \\ 0 & x_s \end{bmatrix} & \succeq & 0
\end{array}
$$

with dual

$$
\begin{array}{llrcl}
 & \max & ny_0 + e^T y & & \\
(5.14) & \text{subject to} & \operatorname{Diag}(y) + y_0(ee^T + I) + S & = & \frac{L}{4} \\
 & & S & \succeq & 0 \\
 & & y_0 & \leq & u
\end{array}
$$

Here x_s is the dual variable corresponding to the upper bound constraint $y_0 \leq u$. Since $I \bullet X = n$, we have $ee^T \bullet X = -x_s$ in (5.13). Similarly the dual variable corresponding to this upper bound constraint in the LP (5.12) should provide an estimate for $ee^T \bullet X$.

5.4. The k equipartition problem. The **k** *equipartition* problem was introduced by Donath and Hoffman [10] and Rendl and Wolkowicz [38]. Our SDP formulation is taken from Lisser and Rendl [30]. It is interesting to note that that for $k = 2$ we have the min bisection problem. The **k** equipartition problem corresponds to a $\{0,1\}$ formulation, and the equivalence of (5.15) for $k = 2$ with the min bisection SDP (5.7) can be easily established.

The Lisser and Rendl [30] formulation is based on the following quadratic program in binary variables.

$$\begin{array}{lrcl} \min & \frac{1}{2}\mathrm{Trace} Z^T L Z & & \\ \text{subject to} & Z^T e & = & le \\ & Ze & = & e \\ & Z_{ij} & = & \{0,1\} \quad \forall i = 1,\ldots,n \quad j = 1,\ldots q, \end{array}$$

L is the Laplacian matrix of the graph, $l = \frac{n}{q}$, where n is the number of nodes and q is the number of partitions desired. Also Z is a $n \times q$, $(0,1)$ matrix whose rows correspond to the vertices V of the graph and whose columns represent the q equal components in which V is to be partitioned. An entry Z_{ij} of this matrix is 1 if vertex i belongs to component j in the partition.

A SDP relaxation is the following [30]

$$\begin{array}{lrcl} \min & \frac{1}{2}\mathrm{Trace} LX & & \\ \text{subject to} & \mathrm{diag}(X) & = & e \\ & Xe & = & le \\ & X & \succeq & 0 \end{array} \tag{5.15}$$

This can also be written as

$$\begin{array}{lrcl} \min & \frac{L}{2} \bullet X & & \\ \text{subject to} & e_i e_i^T \bullet X & = & 1 \quad \forall i = 1,\ldots,n \\ & \frac{1}{2}(e e_i^T + e_i e^T) \bullet X & = & l \quad \forall i = 1,\ldots,n \\ & X & \succeq & 0 \end{array} \tag{5.16}$$

with dual

$$\begin{array}{lrcl} \min & \sum_{i=1}^n y_i & + & l\sum_{i=1}^n y_{n+i} \\ \text{subject to} & S & = & \sum_{i=1}^n e_i e_i^T y_i + \sum_{i=1}^n \frac{1}{2}(e e_i^T + e_i e^T) y_{n+i} + \frac{L}{2} \\ & S & \succeq & 0 \end{array} \tag{5.17}$$

We propose the following d

kEQ1 Setting $d = e_i$, $\forall i = 1,\ldots,n$ gives $y_i + y_{n+i} \geq -\frac{L_{ii}}{2}$, $\forall i$.
kEQ2 Setting $d = e$, gives $\sum_{i=1}^n y_i + n\sum_{i=1}^n y_{n+i} \geq 0$, since $Le = 0$.
kEQ3 Setting $d = (e_i + e_j)$, $\forall \{i,j\} \in E$ gives $y_i + y_j + 2y_{n+i} + 2y_{n+j} \geq -\frac{L_{ii}}{2} - \frac{L_{jj}}{2} - L_{ij}$.
kEQ4 Setting $d = (e_i - e_j)$, $\forall \{i,j\} \in E$ gives $y_i + y_j \geq -\frac{L_{ii}}{2} - \frac{L_{jj}}{2} + L_{ij}$.
kEQ5 The vectors p_i , $i = 1,\ldots,r$, in the bundle P.

The constraints in kEQ2 ensure that the y_{n+i} , $i = 1,\ldots,n$ don't take arbitrarily large negative values, making the LP unbounded.

The resulting LP is

$$
\begin{array}{llrcll}
\min & & \sum_{i=1}^n y_i + l\sum_{i=1}^n y_{n+i} & & & \\
\text{subject to} & & y_i + y_{n+i} & \geq & -\frac{L_{ii}}{2} & \forall i = 1,\ldots,n \\
 & & \sum_{i=1}^n p_{ji}^2 y_i + \sum_{i=1}^n p_{ji}(p_j^T e) y_{n+i} & \geq & -p_j^T \frac{L}{2} p_j & \forall j = 1,\ldots,r \\
(5.18) & & \sum_{i=1}^n y_i + n\sum_{i=1}^n y_{n+i} & \geq & 0 & \\
 & & y_i + y_j + 2y_{n+i} + 2y_{n+j} & \geq & -\frac{L_{ii}}{2} - \frac{L_{jj}}{2} - L_{ij} & \forall \{i,j\} = 1,\ldots,m \\
 & & y_i + y_j & \geq & -\frac{L_{ii}}{2} - \frac{L_{jj}}{2} + L_{ij} & \forall \{i,j\} = 1,\ldots,m
\end{array}
$$

(5.18) has $n + 2m + r + 1$ constraints in all.

5.5. The Lovasz theta function. The Lovasz theta function was introduced by Lovasz [31] in connection with the Shannon capacity of the graph. There are various formulations of the Lovasz theta function, and their equivalence is established in Grotschel, Lovasz and Schrijver [13]. There are also a number of SDP's whose objective value gives the Lovasz theta function. We consider two such formulations in this section. The first formulation is taken from Grotschel, Lovasz and Schrijver [13] and also appears in Gruber and Rendl [14], and the *SDPT3* manual [41].

The Lovasz theta function is an upper bound on the independent set number of a graph, and a lower bound on the chromatic number of the complementary graph. The importance of the Lovasz theta function lies in the fact that computing the independent set number and the chromatic number are *NP complete* problems, whereas the Lovasz theta function, which is sandwiched in between these two numbers, can be computed in polynomial time by solving an SDP. This SDP can be written as

$$
(5.19) \qquad \begin{array}{lrcll}
\min & C \bullet X & & & \\
\text{subject to} & I \bullet X & = & 1, & \\
 & A_k \bullet X & = & 0, & k = 1,\ldots,n \\
 & X & \succeq & 0, &
\end{array}
$$

with dual

$$
(5.20) \qquad \begin{array}{lccccccc}
\min & y_0 & & & & & & \\
\text{subject to} & -y_0 I & + & -\sum_{k=1}^m y_k A_k & + & S & = & C \\
 & & & & & S & \succeq & 0
\end{array}
$$

Here C is the matrix of all minus ones, $A_k = e_i e_j^T + e_j e_i^T$, where the kth edge of the graph (with m edges) is from vertex i to vertex j. Here e_i denotes the ith unit vector. The a in $\mathrm{tr}(X) = a$ is 1, and appears as a constraint in the original SDP.

Since $S = y_0 I + \sum_{k=1}^m y_k A_k + C$ is p.s.d., we require that $d^T S d = d^T(y_0 I + \sum_{k=1}^m y_k A_k + C)d \geq 0, \forall d$.

In particular we propose to use the following d for the Lovasz theta problem.

LT1 Setting $d = e_i$, $i = 1 \ldots n$, gives $y_0 \geq 1$. This constraint implies that a lower bound on the Lovasz theta number is one. (The Lovasz theta number is an upper bound on the maximum clique number of the graph, and each graph trivially has a clique of size 1.)

LT2 Setting $d = e_i + e_j$, $\forall \{i,j\} \in E$ gives $y_0 + y_k \geq 2$, $k = 1,\ldots,m$.

LT3 Setting $d = e_i - e_j$, $\forall \{i,j\} \in E$ gives $y_0 - y_k \geq 0$, $k = 1,\ldots,m$.

LT4 Setting $d = e$ gives $ny_0 + 2\sum_{i=1}^m y_i \geq n^2$.

LT5 The bundle constraints namely the columns p_i , $i = 1,\ldots,r$ of the matrix P.

The constraints in LT2 and LT3 are required to force y_0 to move as y_k move. Otherwise the LP is trivial to solve. The optimal solution is $y_0 = 1$ for an objective value of 1, adjusting the y_k , $k = 1, \ldots, m$ so as to satisfy the remaining constraints.

The resulting LP is

$$\begin{array}{lrcll} \min & y_0 & & & \\ & y_0 + y_k & \geq & 2 & \forall k = 1, \ldots, m \\ & y_0 - y_k & \geq & 0 & \forall k = 1, \ldots, m \\ & ny_0 + 2\sum_{i=1}^m y_i & \geq & n^2 & \\ & p_i^T(y_0 I + \sum_{k=1}^m y_k A_k + C)p_i & \geq & 0 & \forall i = 1, \ldots, r \\ & y_0 & \geq & 1 & \end{array} \tag{5.21}$$

Here $A_k = e_i e_j^T + e_j e_i^T$, $\forall k = \{i, j\}$. (5.21) has $2m + 2 + r$ constraints in all.

There is another SDP relaxation called the maximum stable set relaxation that is equivalent to the Lovasz theta problem. We briefly mention this relaxation below. This equivalence was established in Kleinberg and Goemans [26]. Also refer to Benson and Ye [2] for more details.

The maximum stable set problem is

$$\begin{array}{lrcll} \max & \begin{bmatrix} 0.5 & \ldots & 0 & -0.25 \\ & \ddots & & \vdots \\ 0 & \ldots & 0.5 & -0.25 \\ -0.25 & \ldots & -0.25 & 0 \end{bmatrix} \bullet X & & & \\ \text{subject to} & \operatorname{diag}(X) & = & e & \\ & \frac{1}{2}(e_i e_n^T + e_n e_i^T + e_j e_n^T + e_n e_j^T - e_i e_j^T - e_j e_i^T) \bullet X & = & 1 & \forall \{i,j\} \in E \\ & X & \succeq & 0 & \end{array}$$

with dual

$$\begin{array}{lll} \min & \sum_{i=1}^n y_i & + \quad \sum_{\{i,j\} \in E} y_{ij} \\ \text{subject to} & S & = \\ & \multicolumn{2}{l}{\sum_{i=1}^n e_i e_i^T y_i + \sum_{\{i,j\} \in E} \frac{1}{2}(e_i e_n^T + e_n e_i^T + e_j e_n^T + e_n e_j^T - e_i e_j^T - e_j e_i^T) y_{ij} - C} \\ & S & \succeq \quad 0 \end{array}$$

Here $C = 0.5I - 0.25(ee_n^T + e_n e^T)$. Here the underlying graph G has $n - 1$ vertices, the nth vertex is artificial, with no edges connecting it to the other edges. This vertex is definitely part of the maximal stable set, and is used to identify the stable set in the graph. Moreover the a in $\operatorname{Trace}(X) = a$ is n.

We consider the following d.

MSS1 Setting $d = e_k$, $\forall k = 1, \ldots, n - 1$ gives $y_k \geq 0.5$

MSS2 Setting $d = e_n$ gives $y_n \geq 0$

MSS3 Setting $d = e$ gives $\sum_{i=1}^n y_i + \sum_{\{i,j\} \in E} y_{ij} \geq 0$. This gives a trivial lower bound on the LP objective value.

MSS4 Setting $d = (e_k - e_l)$, $\forall \{k, l\} \in E$ gives $y_k + y_l + y_{kl} \geq 1$.

MSS5 Setting $d = (e_k + e_l)$, $\forall \{k, l\} \in E$ gives $y_k + y_l - y_{kl} \geq 1$.

MSS6 The bundle constraints, i.e. the columns p_k , $k = 1, \ldots, r$ of the matrix P. These give $\sum_{i=1}^n p_{ki}^2 y_i + \sum_{\{i,j\} \in E} (p_{ki} p_{kn} + p_{kj} p_{kn} - p_{ki} p_{kj}) y_{ij} \geq 0.5 - 0.5(p_k^T e)(p_{kn})$.

The resulting LP is

$$
\begin{array}{llcll}
\min & \sum_{i=1}^n y_i + \sum_{\{i,j\}\in E} y_{ij} & & & \\
\text{subject to} & \sum_{i=1}^n y_i + \sum_{\{i,j\}\in E} y_{ij} & \geq & 0 & \\
& y_i + y_j - y_{ij} & \geq & 1 & \forall \{i,j\} \in E \\
& y_i + y_j + y_{ij} & \geq & 1 & \forall \{i,j\} \in E \\
& \sum_{i=1}^n p_{ki}^2 y_i + \sum_{\{i,j\}\in E}(p_{ki}p_{kn} + p_{kj}p_{kn} - p_{ki}p_{kj})y_{ij} & & & \\
& & \geq & 0.5 - 0.5(p_k^T e)p_{kn} & k = 1,\ldots,r \\
& y_i & \geq & 0.5 & i = 1,\ldots,n-1 \\
& y_n & \geq & 0 &
\end{array} \tag{5.22}
$$

(5.22) has $2m + r + 1$ constraints in all.

This problem has a trivial objective value of 0. This is not surprising since our objective was to minimize $\sum_{i=1}^n y_i + \sum_{\{i,j\}\in E} y_{ij}$, whereas constraint MSS3 indicates that this should be greater than 0. We are unable to find any cutting planes that cut off this trivial solution. Note that constraint MSS3 provides a trivial lower bound on the objective function, in the absence of which the problem is unbounded.

5.6. Box constrained QP's. Consider the the following box constrained QP

$$
\begin{array}{lccccc}
\max & x^T Q x & & & & \\
\text{subject to} & -e & \leq & x & \leq & e
\end{array}
$$

A SDP relaxation of this problem, proposed by Ye [46] is

$$
\begin{array}{lrcl}
\max & Q \bullet X & & \\
\text{subject to} & \mathrm{diag}(X) & \leq & e, \\
& X & \succeq & 0,
\end{array} \tag{5.23}
$$

$x_{i,i} + x_{s,i} = 1$

$x_{s,i} \geq 0$

with dual

$$
\begin{array}{lrcl}
\min & e^t y & & \\
\text{subject to} & \mathrm{Diag}(y) & \succeq & Q \\
& y & \geq & 0
\end{array} \tag{5.24}
$$

Note that we can rewrite the primal SDP in the standard form as

$$
\begin{array}{llcll}
\max & \begin{bmatrix} Q & 0 \\ 0 & 0 \end{bmatrix} \bullet \begin{bmatrix} X & 0 \\ 0 & \mathrm{Diag}(x_s) \end{bmatrix} & & & \\
\text{subject to} & A_i \bullet \begin{bmatrix} X & 0 \\ 0 & \mathrm{Diag}(x_s) \end{bmatrix} & = & 1 & \forall i = 1,\ldots,n \\
& \begin{bmatrix} X & 0 \\ 0 & \mathrm{Diag}(x_s) \end{bmatrix} & \succeq & 0 &
\end{array}
$$

Here $A_i \in \mathcal{S}^{2n}$ and is given by $e_i e_i^T + e_{n+i}e_{n+i}^T$.
The dual SDP is given by

$$
\begin{array}{lccccccc}
\min & e^T y & & & & & & \\
\text{subject to} & S & = & \sum_{i=1}^m y_i A_i & + & \begin{bmatrix} -Q & 0 \\ 0 & 0 \end{bmatrix} & \succeq & 0
\end{array}
$$

This problem is similar to the max cut problem, except that all the matrix sizes have been doubled, and the A_i matrices are now the sum of two rank one matrices. The value of $a = n$.

We use the following d.

QPB1 Setting $d = e_i$, $\forall i = 1, \ldots, n$ gives $y_i \geq Q_{ii}$ and $y_i \geq 0$, $\forall i = 1, \ldots, n$. Therefore set $y_i \geq \max\{Q_{ii}, 0\}$. Since Q is a typically a p.s.d. matrix, this constraint is $y_i \geq Q_{ii}$.

QPB2 The columns p_i , $i = 1, \ldots, r$ in the bundle P.

The resulting LP is

$$\begin{array}{lrcll} \min & e^T y & & & \\ \text{subject to} & p_j^T(\sum_{i=1}^m y_i A_i - \begin{bmatrix} Q & 0 \\ 0 & 0 \end{bmatrix})p_j & \geq & 0 & \forall j = 1, \ldots, r \\ & y_i & \geq & Q_{ii} & \end{array} \tag{5.25}$$

(5.25) has $n + r$ constraints in all.

6. A cutting plane LP approach to solving SDP. In this section we describe a scheme to solve SDP, as a sequence of LP's incorporated in a cutting plane framework. The algorithm is outlined in Figure 6.1. For more details, we refer the reader to Krishnan and Mitchell [27].

The objective of this section is twofold.

1. We wish to emphasize the importance of Helmberg's bundle constraints and the box constraints we considered in section 5.
2. We could in practice strengthen the LP relaxations we considered in section 5, by incorporating them in this cutting plane framework. This is especially true for the SDP's with a large number of constraints such as the **k** equipartition problem and the Lovasz theta problem. In this case the initial LP relaxation LDR_1 is the relaxation considered in section 5.

We find that we are able to get tighter relaxations using an interior point code, as compared to the simplex scheme. Moreover since we solve our LP relaxations initially to reasonably high tolerances TOL, we find that it is better to solve the current LP relaxation with a strictly feasible starting point, so as to guarantee primal and dual feasibility, at these tolerances.

6.1. Generating a feasible starting point for LP relaxations. We now illustrate how we generate a strictly feasible starting point for the max cut problem. The technique is similar to the one used in [33]. For convenience we will consider the max cut LP relaxation $LP1$. The primal and dual relaxations are (5.4) and (5.5) respectively.

We generate a strictly feasible starting point for (5.4) by choosing $\mathcal{A}^T y + C$ to be positive definite. This can be done by appropriately increasing y, and checking the positive definiteness of $S = \mathcal{A}^T y + C$ via a Cholesky factorization. This y is strictly feasible for (5.4).

To generate a strictly feasible starting point for (5.5), we observe that we can rewrite (5.5) as

$$\begin{array}{lccc} \max & \sum_{i=1}^n \frac{L_{ii}}{4} x_i + c^T w & & \\ \text{s.t.} & \begin{bmatrix} I & A \end{bmatrix} \begin{bmatrix} x \\ w \end{bmatrix} & = & e \\ & \begin{bmatrix} x \\ w \end{bmatrix} & \geq & 0 \end{array}$$

Cutting plane Algorithm

Input :
$\mathcal{A}(.), b, C$, the optimal solution y_1 to the current relaxation LDR_1.
Parameters :
TOL The starting tolerance to which we solve the LP relaxations. Typically $1e-1$.
s The number of eigenvectors i.e. cutting planes added in each iteration. Typically $s = (20, 30)$.
μ The factor $\in (0, 1)$ by which the tolerance is reduced in each iteration. Typically $\mu = 0.95$.
MAXITER The maximum number of iterations carried out.
for $i = 1$: MAXITER **do**
begin

1. Compute the current dual slack matrix $S = C - \mathcal{A}^t y_i$.
2. Compute the s most negative eigenvalues $\lambda_j, j = 1, \ldots, s$ of S, together with the corresponding normalized eigenvectors $d_j, j = 1, \ldots, s$ using the *Lanczos* scheme. Note that $d_j^T(C - \mathcal{A}^T y)d_j = \lambda_j < 0, j = 1, \ldots, s$.
3. **if** $|\lambda_1| < \epsilon$ or $TOL < 1e{-8}$ **then** exit. Here λ_1 is the most negative eigenvalue.
4. Add these cutting planes $d_j d_j^T \bullet (C - \mathcal{A}^T y) \geq 0, j = 1, \ldots, s$ to the current LP relaxation.
5. Solve LDR_{i+1} using an interior point LP code with a strictly feasible starting point, to the desired tolerance TOL. We illustrate how we generate a strictly feasible starting point for the max cut problem in section 6.1. Let the optimal solution to LDR_{i+1} be y_{i+1}.
6. TOL = $\mu \times$ TOL.
7. We drop constraints if necessary. For more details refer to [27].

end

FIG. 6.1. *The cutting plane algorithm*

Here $x \in \mathbb{R}^n$, $w \in \mathbb{R}^q$ and I,A are the $n \times n$ identity and $n \times q$ matrices respectively.

Set $a_0 = Ae$ i.e. sum all the columns corresponding to w in (5.4).

Thus we have replaced the vector w with a single variable x_0. Now generate a strictly feasible starting point by solving the following LP

$$\begin{array}{lrcl} \max & & & x_0 \\ \text{subject to} & \begin{bmatrix} I & a_0 \end{bmatrix} \begin{bmatrix} x \\ x_0 \end{bmatrix} & = & e \\ & \begin{bmatrix} x \\ x_0 \end{bmatrix} & \geq & 0 \end{array}$$

The optimal solution to this LP is obtained, via a minimum ratio test, by setting $x_0 = \min_{i=1,\ldots,n}(\frac{1}{a_0(i)})$. Hence to generate a strictly feasible starting point we set $x_0 = \mu \min_{i=1,\ldots,n}(\frac{1}{a_0(i)})$ where $\mu \in (0, 1)$ and then assign this value x_0 for all the variables in w and then compute $x_i = 1 - a_0 x_0, i = 1, \ldots, n$. This gives a strictly feasible starting point to (5.5).

7. Computational results. In this section we test the linear programming approach on a number of combinatorial optimization problems, taken from the 7th DI-

MACS Implementation Challenge [37] and Borchers' SDPLIB [4]. For the **k** equipartition problem we take a random instance from Rendl [30] and a 32 node problem from Mitchell [32]. The bundle constraints are computed using Helmberg's spectral bundle code *SBmethod, Version 1.1* [19] available at *http://www.zib.de/helmberg/index.html*. *CPLEX 6.5* [24] is employed in solving the LP relaxations, within the *MATLAB* framework. All tests are executed on a *Sun Ultra 5.6, 440MHz* machine with *128* MB of memory.

The spectral bundle approach requires a number of parameters. We briefly mention some of the important parameters together with their default values. For a detailed description of these parameters, we refer the reader to Helmberg's user's manual for *SBmethod* [19].

1. The relative tolerance (-*te*). The default value is $1e-5$.
2. The size of the bundle i.e. the number of columns in P. This in turn is controlled by
   (a) The maximum number of vectors kept n_k (-*mk*). The default value is 20.
   (b) The maximum number of vectors added (-*ma*). The default value is 5.
   (c) The minimum number of vectors added n_{min} (-*mik*). The default value is 5.
3. The time limit in seconds (-*tl*).

It must be emphasized that the main computational task in the bundle approach is computing the vectors in the bundle P. Solving the resulting LP relaxations is relatively trivial.

For some of the problems the bundle approach failed to converge to the desired tolerance ($1e-5$). For these problems, we provide run times in parentheses next to the bundle size r , and also the objective value attained by the bundle approach next to the optimal SDP objective value.

The columns in the tables represent

n Problem size i.e. the number of nodes in the graph.
k Number of SDP constraints.
m Number of edges in the graph.
r Bundle size, the number of columns in P.
% Error $|\frac{SDP-LP}{SDP} \times 100|$.
Table 7.1 SDP LP1 LP2 The objective value of the various relaxations.
  % Error $\frac{SDP-LP2}{SDP} \times 100$.
Table 7.2 m1 m2 Number of constraints in LP's (5.4) and (5.6) respectively.
Table 7.3 m1 Number of constraints in LP (5.12).
Table 7.4 q Number of equipartitions required.
  m1 Number of constraints in LP (5.18).
Table 7.5 m1 Number of constraints in LP (5.21).
Table 7.6 m1 Number of constraints in LP (5.25).
Table 7.7 Bundle LP The objective value of LP relaxation (5.6).
  Interior LDR The objective value of LDR in the interior point cutting plane approach.
  Interior LPR The objective value of LPR in the interior point cutting plane approach.
  Eig The most negative eigenvalue of the dual slack matrix S at the end of Iter iterations.
  Iter The number of iterations carried out.

Table 7.8 m_{sb} The number of constraints in the LP relaxation (5.6).
m_{icp} The number of constraints in LDR in the cutting plane approach.

Name	n	m	r (No of hrs)	SDP	LP1	LP2	% Error
toruspm-8-50 [1]	512	1536	16	527.81	525.91	526.37	0.27
toruspm3-15-50 [1]	3375	10125	21	3.47e+03	3.43e+03	3.45e+03	0.81
torusg3-8 [1]	512	1536	13	4.57e+07	4.54e+07	4.55e+07	0.43
torusg3-15 [1]	3375	10125	18	3.13e+08	3.10e+08	3.11e+08	0.69
mcp100 [2]	100	269	10	226.16	225.75	225.86	0.13
mcp124-1 [2]	124	149	20	141.99	141.06	141.95	0.03
mcp124-2 [2]	124	318	11	269.88	268.97	269.21	0.25
mcp124-3 [2]	124	620	11	467.75	467.37	467.44	0.06
mcp124-4 [2]	124	1271	10	864.41	863.72	863.81	0.07
mcp250-1 [2]	250	331	10	317.26	317.18	317.21	0.02
mcp250-2 [2]	250	612	14	531.93	531.18	531.38	0.1
mcp250-3 [2]	250	1283	13	981.17	980.32	980.48	0.07
mcp250-4 [2]	250	2421	13	1681.96	1679.70	1680.00	0.12
mcp500-1 [2]	500	625	10	598.15	594.12	596.67	0.25
mcp500-2 [2]	500	1223	13	1070.06	1069.90	1069.95	0.01
mcp500-3 [2]	500	2355	14	1847.97	1843.20	1844.13	0.21
mcp500-4 [2]	500	5120	15	3566.74	3559.80	3560.64	0.17
maxG11 [2]	800	1600	11	629.16	625.99	626.99	0.35
maxG32 [2]	2000	4000	14	1567.64	1557.91	1560.94	0.43
maxG51 [2]	1000	5909	19	4003.81	3988.30	3992.60	0.3
maxG55 [2]	5000	14997	25 (10)	12870	11523.66	12855.03	0.12
maxG60 [2]	7000	17148	21 (1)	15222.27 [6] (15318.40)	8606.60	13837.29	9.1
maxG60 [2]	7000	17148	24 (2)	15222.27 [6] (15298.20)	8615.31	14286.71	6.15
maxG60 [2]	7000	17148	25 (6)	15222.27 [6] (15270.29)	5334.17	14796.33	2.79

Table 7.1
Max Cut Test Results

In table 7.1 we compare the SDP objective value with the LP relaxations $LP1$ (5.4) and $LP2$ (5.6). It is seen that in most cases $LP1$ alone provides a fairly good approximation to the SDP objective value. However for problems *mcp124-1*, *mcp500-1*, *maxG11*, *maxG32* we need to work with the relaxation $LP2$. These examples underline the importance of the m rank 2 constraints in $LP2$. The LP relaxation $LP2$ provides an excellent approximation to the SDP with the $\%error$ well below 1% of the SDP objective value.

The bundle approach fails to converge for problems *maxG55* and *maxG60*. For *maxG55* we run the bundle code for 10 hours. The bundle approach attains an objective value of 12870 (reported SDP objective in SDPLIB is 9999.210). Moreover since the LP relaxations $LP1$ and $LP2$ are larger than 9999.210 it appears that the reported solution in SDPLIB is incorrect. For the *maxG60* problem we consider three

[1]DIMACS [37]
[2]SDPLIB [4]
[3]Rendl [30]
[4]Mitchell [32]
[5]Memory exceeded
[6]Spectral bundle failed to terminate

runs with the bundle code, of durations 1 hour, 2 hours and 6 hours respectively. It is seen that longer the run, the closer the bundle objective to the SDP objective value. The $LP2$ relaxation is also tighter.

Name	SDP		LP1	LP2
	k	n	m1	m2
toruspm-8-50	512	512	528	2064
toruspm3-15-50	3375	3375	3396	13521
torusg3-8	512	512	525	2061
torusg3-15	3375	3375	3393	13518
mcp100	100	100	110	379
mcp124-1	124	124	144	293
mcp124-2	124	124	135	453
mcp124-3	124	124	135	755
mcp124-4	124	124	134	1405
mcp250-1	250	250	260	591
mcp250-2	250	250	264	876
mcp250-3	250	250	263	1546
mcp250-4	250	250	263	2684
mcp500-1	500	500	510	1135
mcp500-2	500	500	513	1736
mcp500-3	500	500	514	2869
mcp500-4	500	500	515	5635
maxG11	800	800	811	2411
maxG32	2000	2000	2014	6014
maxG51	1000	1000	1019	6928
maxG55	5000	5000	5025	20022
maxG60	7000	7000	7025	24173

TABLE 7.2
Sizes of the max cut relaxations

We list the sizes of the three max cut relaxations in table 7.2. The SDP has n constraints, whereas $LP1$ and $LP2$ are LP relaxations with approximately $(n+\sqrt{n}) = O(n)$ and $(n+m+\sqrt{k}) = O(n^2)$ constraints respectively.

In table 7.3 we compare the SDP objective value with the LP relaxation (5.12). Here $u = 1$ is the upper bound on the variable y_0. It is seen that the LP relaxation provides an excellent approximation to the SDP objective value. The value of the dual variable, corresponding to the upper bound constraint $y_0 \leq 1$ provides an estimate for $|ee^t \bullet X|$. This value is below 0.1 for all but one of the reported instances. A typical LP relaxation has approximately $(m+n+\sqrt{k}) = O(n^2)$ constraints, where $k = n+1$.

In table 7.4 we compare the SDP objective value of the **k** equipartition problem with the value of LP relaxation (5.18). It is interesting to note that the LP relaxation of the bisection problem, i.e. **k** equipartition with $q = 2$, does not have an unbounded optimal face as the LP relaxation $LP1$ (5.11) of min bisection. However the SDP relaxation (5.15) which represents a $\{0,1\}$ formulation of min bisection, as opposed to the $\{-1,1\}$ SDP formulation (5.7) is relatively harder to solve, using the bundle approach, since we are dealing with more constraints. Among **k** equipartition problems, we need a larger bundle P for the bipartition problems *gpp124-1*, *gpp250-1*. For these problems we choose the maximum number of vectors $n_k = 50$ and $n_{min} = 25$. With

Name	m	n	r	SDP	LP	m1	% Error	$\lvert ee^T \bullet X\rvert$
bm1 [1]	4711	882	10	23.44	24.99	10315	6.59	0.06
gpp100 [2]	264	100	10	44.94	45.14	639	0.44	0.02
gpp124-1 [2]	149	124	10	7.34	7.34	433	0.01	0.00
gpp124-2 [2]	318	124	10	46.86	47.27	771	0.55	0.03
gpp124-3 [3]	620	124	11	153.01	153.40	1376	0.16	0.01
gpp124-4 [2]	1271	124	11	418.99	419.98	2678	0.14	0.04
gpp250-1 [2]	331	250	10	15.45	15.45	923	0.88	0.00
gpp250-2 [2]	612	250	12	81.87	82.09	1487	0.26	0.02
gpp250-3 [2]	1283	250	13	303.50	305.21	2830	0.56	0.02
gpp250-4 [2]	2421	250	13	747.30	750.58	5106	0.43	0.03
gpp500-1 [2]	625	500	11	25.30	26.25	1762	3.62	0.18
gpp500-2 [2]	1223	500	13	156.06	157.73	2960	1.06	0.06
gpp500-3 [2]	2355	500	15	513.02	517.91	5226	0.95	0.02
gpp500-4 [2]	5120	500	15	1567.02	1575.95	10756	0.57	0.01
biomedP [1]	629839	6514	-	33.60	MM [5]	-	-	-
industry2 [1]	798219	12637	-	65.61	MM [5]	-	-	-

TABLE 7.3
Min Bisection Test Results

Name	m	n	q	r (No of Hrs)	SDP	LP	m1	% Error
nfl1 [4]	496	32	8	9	7.34e+05	7.32e+05	1034	0.23
nfl2 [4]	496	32	4	9	6.29e+05	6.29e+05	1034	0
A100 [3]	4070	100	2	10	8.62e+05	8.62e+05	8251	0.00
gpp100 [2]	364	100	2	10	44.94	44.94	839	0.00
gpp124-1 [2]	149	124	2	10	7.34	43.04	433	486.09
gpp124-1 [2]	149	124	2	31	7.34	7.34	454	0.01
gpp124-2 [2]	318	124	2	8	46.86	46.86	769	0.00
gpp250-1 [2]	331	250	2	40 (3)	15.45 [6] (15.41)	18.76	953	21.44
gpp250-2 [2]	612	250	2	11	81.87	81.87	1486	0.00

TABLE 7.4
k Equipartition Test Results

these enhanced bundle sizes we are able to get tighter LP approximation for *gpp1241*. With the default bundle size we are only able to get an LP approximation of 43.0370 (SDP objective value is 7.3431), but with the larger bundle size our LP relaxation is now 7.3443 for a $\%error$ of 0.016. The larger bundle sizes however result in r much larger than $\sqrt{k}$. A typical LP relaxation has approximately $(m + n + \sqrt{k}) = O(n^2)$ constraints, where $k = 2n$.

In table 7.5 we compare the SDP objective value of the Lovasz theta problem with the value of the LP relaxation (5.21). Since the number of constraints in the Lovasz theta problem is $O(m)$ which could as large as $O(n^2)$, a larger bundle is typically required. Thus the bundle method is fairly time consuming on these problems. For instance *theta3* the bundle approach requires about 4 hours to converge to the desired tolerance, whereas it fails to converge for instances *theta4*, *theta5*, *theta6*. For these problems we utilize the bundles after a 6 hour run. The objective value attained by the bundle approach during this period, is reported in parentheses next to the SDP

Name	m	n	r (No of Hrs)	SDP	LP	m1	% Error
hamming-9-8 [1]	2304	512	23	224	221.68	4633	1.03
hamming-7-5-6 [1]	1792	128	12	42.67	42.53	3598	0.33
hamming-10-2 [1]	23040	1024	-	102.4	MM [5]	-	-
hamming-11-2 [1]	56320	2048	-	170.67	MM [5]	-	-
hamming-8-3-4 [1]	16128	256	43	25.6	25.32	32301	1.09
hamming-9-5-6 [1]	53760	512	-	85.33	MM [5]	-	-
theta1 [2]	103	50	10	23	22.95	218	0.22
theta2 [2]	497	100	20	32.87	32.84	1016	0.10
theta3 [2]	1105	150	25	42.17	40.14	2237	4.81
theta3 [2]	1105	150	47	42.17	42.11	2259	0.14
theta4 [2]	1948	200	25(6)	50.32 [6] (50.35)	38.12	3923	24.25
theta4 [2]	1948	200	51(6)	50.32 [6] (50.33)	49.63	3949	1.38
theta5 [2]	3027	250	25(6)	57.23 [6] (57.28)	33.30	6081	41.81
theta5 [2]	3027	250	51(6)	57.23 [6] (57.58)	51.87	6107	9.38
theta5 [2]	3027	250	51(10)	57.23 [6] (57.39)	54.64	6107	4.79
theta6 [2]	4374	300	25(6)	63.48 [6] (63.56)	30.06	8775	52.64
theta6 [2]	4374	300	59(6)	63.48 [6] (65.05)	50.54	8809	20.39
theta6 [2]	4374	300	58(10)	63.48 [6] (64.32)	54.57	8808	14.03
theta6 [2]	4374	300	58(15)	63.48 [6] (63.91)	57.27	8808	9.78

Table 7.5
Lovasz theta Test Results

objective value. We retain the default bundle parameters. However for the DIMACS *Hamming* instances, since the optimal solution can be computed analytically, the bundle approach is able to find the optimal bundle in a few iterations. We are unable to solve to solve *hamming-10-2*, *hamming-11-2* and *hamming9-5-6* where we run out of memory. It is seen that in the instances where the bundle approach converges, our LP solution is an excellent approximation to the SDP objective value with the *%error* under 1%. To improve on the LP relaxations for the SDPLIB problems *theta4*, *theta5* and *theta6*, we choose larger bundle sizes $n_k = 50$ and $n_{min} = 25$ and $n_{add} = 25$. However this slows down the bundle code. We compute the LP relaxations, choosing the bundle after 6 and 10 hour runs respectively. We are able to considerably strengthen our LP relaxation in all cases. A typical LP relaxation has approximately $(m + \sqrt{k}) = O(n^2)$ constraints, where $k = O(m)$. It must be emphasized here that all the SDPLIB Lovasz theta problems can be solved in under an hour using any interior point package such as [41].

Name	k	n	r	SDP	LP	m1	% Error
qpG11 [2]	800	1600	11	2448.66	2434.50	811	0.57
qpG51 [2]	1000	2000	10	1.18e+04	1.18e+04	1010	0.39

Table 7.6
Box QP Test Results

Table 7.6 compares the SDP objective of the box constrained QP with the value of the LP relaxation (5.25). This LP provides an excellent approximation to the SDP objective value and has approximately $(n + \sqrt{k}) = O(n)$ constraints, where $k = n$.

Name	SDP	Bundle LP	Interior LDR	Interior LPR	Eig	Iter
mcp100 [2]	226.16	225.85	226.13	226.11	-1.1e-3	200
mcp124-1 [2]	141.99	141.95	141.94	141.88	-1.6e-3	185
mcp124-2 [2]	269.88	269.21	269.76	269.72	-2.8e-3	200
mcp124-3 [2]	467.75	467.44	467.40	467.35	-1.2e-2	200
mcp124-4 [2]	864.41	863.81	862.69	862.67	-3.7e-2	200
mcp250-1 [2]	317.26	317.21	316.95	316.91	-1.1e-2	200
mcp250-2 [2]	531.93	531.38	529.77	529.73	-2.7e-2	200
mcp250-3 [2]	981.17	980.48	976.98	976.92	-5.3e-2	200
mcp250-4 [2]	1681.96	1680.00	1675.23	1675.14	-8.5e-2	200
mcp500-1 [2]	598.15	596.67	594.28	586.68	-3.5e-2	100
mcp500-2 [2]	1070.06	1069.95	1059.51	1049.32	-5.3e-2	100
mcp500-3 [2]	1847.97	1844.13	1830.11	1823.00	-1.2e-1	100
mcp500-4 [2]	3566.74	3560.64	3534.11	3532.74	-1.8e-1	140
maxG11 [1]	629.16	626.99	627.31	619.98	-1.8e-2	100
toruspm-8-50 [1]	527.81	525.91	516.90	508.87	-6e-2	100

TABLE 7.7
Strengths of various LP relaxations on max cut

Name	m_{sb}	m_{icp}
mcp100 [2]	379	869
mcp124-1 [2]	293	718
mcp124-2 [2]	453	1026
mcp124-3 [2]	755	1221
mcp124-4 [2]	1405	1332
mcp250-1 [2]	591	1224
mcp250-2 [2]	876	1911
mcp250-3 [2]	1546	2312
mcp250-4 [2]	2684	2465
mcp500-1 [2]	1135	1626
mcp500-2 [2]	1736	1929
mcp500-3 [2]	2869	2253
mcp500-4 [2]	5635	3195
maxG11 [1]	2411	2610
toruspm-8-50 [1]	2064	2197

TABLE 7.8
Sizes of the various LP relaxations on max cut

Table 7.7 compares the LP relaxation with the bundle and box constraints discussed in section 5 which we will henceforth refer to as the bundle LP, with the cutting plane LP relaxation introduced in section 6 for the max cut problem. For the cutting plane approach the initial LP relaxation LDR_1 is obtained by setting $y_i = \frac{L_{ii}}{4}$, $\forall i = 1, \ldots, n$. The other cutting plane parameters discussed in section 6 are $TOL = 1$, $s = 20$, $\mu = 0.95$ and $MAXITER = 100$. We used Zhang's *LIPSOL* [47], with a strictly feasible starting point to solve the sequence of interior LP's. Since we end up with a tolerance TOL of $(0.95)^{100} = 5e-3$ at the end of 100 iterations and the primal and dual objective values need not necessarily agree, we provide the objective value of the dual (LPR), which is always a lower bound on the SDP objective value. It is seen that the bundle LP provides a tighter relaxation than the interior LP for most instances, and is also the smaller LP relaxation. However in some of the smaller test cases such as *mcp124-1*, we are able to do better with the interior point code. The fact that the interior point LP is giving good SDP relaxations is reflected in the very small negative eigenvalues of the dual slack matrix S, that we encounter. For most cases this value is smaller than 0.1 in absolute value.

We also tried incorporating the cutting plane framework, where the sequence of LP's are solved using the *CPLEX 6.5* [24] simplex solver. The interior cutting plane LP relaxation clearly outperforms the simplex LP relaxation, and we are able to generate better cutting planes using the former approach. Moreover since we are solving the LP's initially to high tolerances TOL in the interior approach, we seem to get fairly quickly to the periphery of the positive semidefinite cone, reflected in the small negative eigenvalues we encounter. As a result we add not only stronger, but fewer cutting planes in the interior point cutting plane scheme as compared to the simplex approach, with the result that the intermediate LP relaxations can be solved quickly.

The main objective of table 7.7 is to emphasize that the bundle together with the box constraints introduced in section 5 are optimal in a certain sense. To make the interior cutting plane approach competitive with the bundle LP, further refinements are necessary. We pursue this in Krishnan and Mitchell [27].

8. Conclusions. We have presented an LP approach to solving semidefinite programming problems. This approach requires the bundle constraints generated by the *spectral bundle* approach due to Helmberg and Rendl. The number of these constraints is bounded by the square root of the number of constraints k in the SDP. Typically fewer than these are required, due to the aggregation employed in the bundle approach. In addition to the bundle constraints, a few others *polynomial* in the problem size n are generally required. The main computational task in the LP approach is in computing the optimal bundle P. Solving the resulting LP relaxations is relatively trivial. We have also presented an interior cutting plane framework, which could be used to solve SDP as a sequence of LP's. We use this cutting plane approach to demonstrate the importance of the bundle constraints in the LP formulation.

It appears from the results in the paper that

- The LP approach is very successful in solving the max cut and the box constrained QP problems. A typical maxcut LP requires approximately $(n + m + \sqrt{k})$ constraints, where $k = n$. For the box constrained QP instances reported, we find that around $(n + \sqrt{k})$ constraints suffice. These LP's can be solved easily using any of the commercial packages available.
- The original LP relaxation of the min bisection problem (5.11), like the SDP (5.7) has an unbounded optimal face. Hence we need to introduce an additional constraint $y_0 \leq u$ in our LP relaxation. Here y_0 is the dual variable corresponding to the equipartition constraint $ee^T \bullet X = 0$. Thus we are in practice, solving a variant of the min bisection problem. However on all the reported problems the value $|ee^T \bullet X|$ is very small of the order of 0.1. A typical LP has approximately $(m + n + \sqrt{k})$ constraints, where $k = n + 1$.
- The spectral bundle approach is fairly time consuming on Lovasz theta problems. This is because the number of constraints in the SDP is $O(m)$. Moreover we need larger bundle sizes than those provided by the bundle approach to tighten our LP relaxations. It appears that the traditional interior point

methods outperform the spectral bundle method, especially on some of the smaller Lovasz theta problems. A typical LP has approximately $(m + \sqrt{k})$ constraints, where $k = m + 1$.

- The **k** equipartition SDP (5.15) does not have an unbounded optimal face, unlike the min bisection SDP (5.7). However (5.15) has more constraints k and hence computing the optimal bundle is time consuming. A typical LP has approximately $(m + n + \sqrt{k})$ constraints where $k = 2n$. In certain instances, we need larger bundle sizes $r > \sqrt{k}$ to tighten our LP relaxations.
- For the maximum stable set formulation of the Lovasz theta number, we are unable to find cutting planes, that cut off the point with a trivial objective value zero. We intend to investigate, incorporating the bundle LP within the cutting plane framework of section 6, and try to cut off this point. Also we could in practice strengthen the bundle LP relaxations, by incorporating it within the cutting plane framework. We investigate this in greater detail in Krishnan and Mitchell [27].
- The bundle LP, with the additional box constraints introduced in section 5, is superior to the naive interior point cutting plane approach discussed in section 6. Not only are we are able to get tighter relaxations in the former case, but the resulting LP's are smaller as well. Thus the bundle constraints seem to be optimal in a certain sense, in that they identify an *important subspace* on which the matrix of dual slacks S should be positive semidefinite, and directions not in subspace do not seem all that important.
- The spectral bundle approach depends on a number of parameters, especially the maximum and minimum bundle sizes, the number of Lanczos vectors added in each iteration. In this paper we have experimented with various bundle sizes r, especially with regard to the Lovasz theta and **k** equipartition problems. Further investigation of the choice of these parameters, especially the role they play with regard to the strength of our LP relaxations is necessary.
- Another interesting idea is in trying to estimate how much P in the spectral bundle scheme changes in two consecutive iterations. Since in the restricted eigenvalue problem (3.2) we have a quadratic regularization term, which penalizes us from going too far from the current iterate, we could in practice utilize some of the intermediate bundles P in tightening our LP relaxations. This is especially true for SDP's with a large number of constraints such as the Lovasz theta problem. We intend to investigate this in a future paper.

To conclude it is felt that a beginning is made to solve an SDP, with a constant trace on the primal feasible set, as an LP. Although SDP's are *semi infinite* LP's, we provide empirical evidence that only a few constraints, polynomial in the problem size n, are typically required. Furthermore one could incorporate the above framework in a cutting plane LP approach, to solving semidefinite programs. We pursue this in [27].

9. Acknowledgements. The authors wish to thank Christoph Helmberg for making his spectral bundle code available to us and for useful discussions.

REFERENCES

[1] Farid Alizadeh, *Interior Point Methods in Semidefinite Programming with applications to Combinatorial Optimization*, SIAM Journal on Optimization, 5 (1995), pp. 13-51.

[2] S. Benson and Yinyu Ye, *Approximating Maximum Stable Set and Minimum Graph Coloring Problems with the Positive Semidefinite Relaxation*, Working Paper, University of Iowa, Iowa, 1999.

[3] S.J. Benson, Yinyu Ye and Xiong Zhang, *Solving Large-Scale Semidefinite Programs for Combinatorial Optimization*, SIAM Journal on Optimization, 10 (2000), pp. 443-461.

[4] Brian Borchers, *SDPLIB 1.2, A Library for Semidefinite Programming Test Problems*, Optimization Methods and Software, 11(1999), pp. 683-690.

[5] S. Burer and R.D.C. Monteiro, *A nonlinear programming algorithm for solving semidefinite programs via low-rank factorization*, Working paper, School of ISyE, Georgia Tech., Atlanta, GA, March 2001.

[6] S. Burer, R.D.C. Monteiro and Y. Zhang, *Solving Semidefinite Programs via Nonlinear Programming. Part I : Transformations and Derivatives*, Working paper, School of ISyE, Georgia Tech., Atlanta, GA, September 1999.

[7] S. Burer, R.D.C. Monteiro and Y. Zhang, *Solving Semidefinite Programs via Nonlinear Programming. Part II : Interior point Methods for a Subclass of SDP's*, Working paper, School of ISyE, Georgia Tech., Atlanta, GA, October 1999.

[8] S. Burer, R.D.C. Monteiro and Y. Zhang, *Rank-Two Relaxation Heuristics for Max-Cut and Other Binary Quadratic Programs*, Working paper, School of ISyE, Georgia Tech., Atlanta, GA, November 2000.

[9] S. Burer, R.D.C. Monteiro and Y. Zhang, *Maximum stable set formulations and heuristics based on continuous optimization*, Working paper, School of ISyE, Georgia Tech., Atlanta, GA, December 2000.

[10] W.E. Donath and A.J. Hoffman, *Lower bounds for the partitioning of graphs*, IBM Jl. of Research and Development 17(1973), pp. 120-125.

[11] Alan Frieze and M.R. Jerrum, *Improved approximation algorithms for Max k-Cut and Max Bisection*, Algorithmica 18, pp. 61-77.

[12] M. Goemans and D.P. Williamson, *Improved approximation algorithms for max cut and satisfiability problems using semidefinite programming*, J. A.C.M 42 (1995), pp. 1115-1145.

[13] M. Grotschel, L. Lovasz and A. Schrijver, *Geometric Algorithms and Combinatorial Optimization*, Springer-Verlag, New York, 1988.

[14] G. Gruber and F. Rendl, *Computational experience with Stable Set Relaxations*, Department of Mathematics, University of Klagenfurt, Austria, December 2000.

[15] Christoph Helmberg, *Semidefinite Programming*, ZIB-Report ZR-00-34, Konrad-Zuse-Zentrum Berlin, October 2000.

[16] C. Helmberg and F. Rendl, *A Spectral Bundle Method for Semidefinite Programming*, SIAM Journal on Optimization, 10 (2000), pp. 673-696.

[17] C. Helmberg and K.C. Kiwiel, *A Spectral Bundle Method with bounds*, ZIB Preprint SC-99-37, Konrad-Zuse-Zentrum Berlin, December 1999.

[18] C. Helmberg and F. Oustry, *Bundle methods to minimize the maximum eigenvalue function.* In : H. Wolkowicz, R. Saigal, and L. Vandenberghe, eds., *Handbook of Semidefinite Programming*, Kluwer Academic Publishers, Boston-Dordrect-London, 2000, pp. 307-337.

[19] Christoph Helmberg, *SBmethod : A C++ Implementation of the Spectral Bundle Method*, ZIB-Report ZR-00-35, Konrad -Zuse-Zentrum Berlin, October 2000.

[20] R. Hettich and K.O. Kortanek, *Semi-Infinite Programming : Theory, Methods, and Applications*, SIAM Review 35(3) (1993), pp. 380-429.

[21] Roger Horn and Charles Johnson, *Matrix Analysis*, Cambridge University Press, 1990.

[22] J.B. Hiriart Urruty and C. Lemarechal, *Convex Analysis and Minimization Algorithms I*, Springer Verlag, Berlin, Heidelberg, 1993.

[23] J.B. Hiriart Urruty and C. Lemarechal, *Convex Analysis and Minimization Algorithms II*, Springer Verlag, Berlin Heidelberg, 1993.

[24] ILOG CPLEX 6.5, *ILOG Inc.*, 1080 Linda Vista Avenue, Mountain View, CA 94043.

[25] K.C. Kiwiel, *Proximity Control in bundle methods for convex nondifferentiable minimization*, Mathematical Programming 46 (1990), pp. 105-122.

[26] Jon Kleinberg and Michel Goemans, *The Lovasz Theta Function and a Semidefinite Programming Relaxation of Vertex Cover*, SIAM Journal on Discrete Mathematics, 11 (1998), pp. 196-204.

[27] Kartik Krishnan and John E. Mitchell, *A cutting plane LP approach to solving semidefinite programming problems*, in preparation, Dept. of Mathematical Sciences, RPI, Troy, NY, May 2001.

[28] C. Lemarechal, *Nondifferentiable Optimization.* In : G.L. Nemhauser, A.H.G. Rinnooy Kan, and M.J. Todd, eds., *Optimization*, North-Holland, Amsterdam-New York-Oxford-Tokyo, 1989, pp. 529-572.

[29] A.S. Lewis and M.L. Overton, *Eigenvalue Optimization*, Acta Numerica 5(1996), pp. 149-190.

[30] A. Lisser and F. Rendl, *Telecommunication Clustering using Linear and Semidefinite Programming*, Dept. of Mathematics, University of Klagenfurt, Austria, December 2000.

[31] L. Lovasz, *On the Shannon capacity of a graph*, IEEE Transactions on Information Theory, 25(1979), pp. 1-7.

[32] John E. Mitchell, *Realignment in the NFL*, Dept. of Mathematical Sciences, RPI, Troy, NY, November 2000.

[33] J.E. Mitchell and M.J. Todd, *Solving combinatorial optimization problems using Karmarkar's algorithm*, Mathematical Programming, 56(1992), pp. 245-282.

[34] Y. Nesterov and A. Nemirovskii, *Interior Point Polynomial Algorithms in Convex Programming*, in SIAM Studies in Applied Mathematics, Philadelphia, PA, 1993.

[35] Francois Oustry, *A second order bundle method to minimize the maximum eigenvalue function*, Mathematical Programming 89 (2000), pp. 1-33.

[36] Gabor Pataki, *On the Rank of Extreme Matrices in Semidefinite Programs and the Multiplicity of Optimal Eigenvalues*, Mathematics of Operations Research 23(2), pp. 339-358.

[37] Gabor Pataki and Stefan Schmieta, *The DIMACS library of semidefinite -quadratic-linear programs*, Computational Optimization Research, Columbia University, NY, November 1999.

[38] F. Rendl and H. Wolkowicz, *A Projection technique for partitioning the nodes of a graph*, Annals of Operations Research 58(1995), pp. 155-179.

[39] H. Schramm and J. Zowe, *A version of the bundle idea for minimizing a nonsmooth function : Conceptual idea, convergence analysis, numerical results*, SIAM Journal on Optimization 2 (1992), pp. 121-152.

[40] M.J. Todd, *Semidefinite Optimization*, Survey paper, Department of Operations Research, Cornell University, NY, February 2001

[41] K.C. Toh, M.J. Todd and R.H. Tutuncu, *SDPT3 - A Matlab Software Package for semidefinite programming, version 2.1*, Optimization Methods and Software, 11 (1999), pp. 1-49.

[42] L. Vandenberghe and S. Boyd, *Semidefinite Programming*, SIAM Review, 38(1996), pp. 49-95.

[43] R. Vanderbei and H. Benson, *On Formulating Semidefinite Programming Problems as Smooth Convex Nonlinear Optimization Problems*, ORFE-99-01, Princeton University, NJ, January 2000.

[44] Henry Wolkowicz, Romesh Saigal and Lieven Vandenberghe, *Handbook on Semidefinite Programming*, Kluwer Academic Publishers, 2000.

[45] Yinyu Ye, *A .699 Approximation Algorithm for Max Bisection*, Mathematical Programming, 90(1) (2001), pp. 101-111.

[46] Yinyu Ye, *Approximating quadratic programming with bound and quadratic constraints*, Mathematical Programming, 84 (1999), pp. 219-226.

[47] Yin Zhang, *Solving Large-Scale Linear Programs by Interior Point Methods Under the MATLAB Environment*, Technical Report, Department of Computational and Applied Mathematics, Rice University, Houston, Texas, 1997, pp. 1-26.