Cutting plane methods for semidefinite programming *

Kartik Krishnan (kartik@caam.rice.edu)
Department of Computational and Applied Mathematics, Rice University, 6100 Main Street, Houston, Texas, 77005

John E. Mitchell (mitchj@rpi.edu)
Department of Mathematical Sciences, Rensselaer Polytechnic Institute, 110 8th Street, Troy, New York, 12180. http://www.rpi.edu/~mitchj Corresponding author.

Abstract. Interior point methods for semidefinite programming (SDP) are fairly limited in the size of problems they can handle. Cutting plane methods provide a means to solve large scale SDP's cheaply and quickly. We give a survey of various cutting plane approaches for SDP in this paper. These cutting plane approaches arise from two perspectives: the first is based on the polynomial separation oracle for the SDP that is utilized by polynomial interior point cutting plane methods; the second rewrites an SDP with a bounded feasible set as an eigenvalue optimization problem, which in turn is solved using bundle methods for nondifferentiable optimization.

We present an accessible and unified introduction to various cutting plane approaches that have appeared in the literature; in particular we show how each approach arises as a natural enhancement of a primordial LP cutting plane scheme based on a semi-infinite formulation of the SDP.

Keywords: Semidefinite programming, nondifferentiable optimization, cutting plane methods.

1. Introduction

Semidefinite Programming (SDP) has been one of the most exciting and active areas in optimization recently. This tremendous activity was spurred by the discovery of efficient interior point algorithms for solving SDP, and important applications of the SDP in control, developing approximation algorithms for combinatorial optimization problems, finance, statistics etc. Some excellent references for SDP include the website maintained by Helmberg (1996), the survey papers by Todd (2001) and Vandenberghe and Boyd (1996), and the SDP handbook edited by Wolkowicz et al. (2000). However these applications require effective techniques for solving the SDP. Although interior point algorithms are a great theoretical tool, and offer worst case polynomial complexity they are fairly limited in the size of problems they can handle. We investigate cutting plane approaches for the SDP in this paper with a view to solve large scale semidefinite programming problems.

* This work was supported in part by NSF grant number CCR–9901822

Our aim is also to tie together all the cutting plane approaches for the SDP which have appeared in the literature so far.

The cutting plane approaches for SDP fall in two categories:

- **Polynomial interior point cutting plane methods:** These methods are based on the equivalence of separation and optimization as established in Grötschel et al. (1993). The SDP has a polynomial time separation oracle, and hence it can be solved within this framework. Good surveys of such methods appear in Goffin and Vial (2002) and Mitchell (2001). We must emphasize that we could use these methods to solve the SDP in polynomial time, and that these complexities compare well with interior point methods for SDP (see Krishnan (2002) for more details).

- **Bundle methods for nondifferentiable optimization:** An SDP with some additional restrictions (see Assumption 2) can be written as an eigenvalue optimization problem. These are convex but non-smooth optimization problems, that can be handled by bundle methods for nondifferentiable optimization. A good survey on bundle methods appear in Lemarechal (1989). Polynomial time complexity proofs for these methods are not known, but the spectral bundle method (Helmberg and Rendl, 2000) appears to be very efficient in practice.

We are aware of at least three distinct cutting plane approaches for the SDP namely the spectral bundle method due to Helmberg (2000), Helmberg and Kiwiel (1999), Helmberg and Oustry (2000), Helmberg and Rendl (2000), and Oustry (2000), an LP cutting plane scheme for the SDP due to Krishnan (2002) and Krishnan and Mitchell (2001), which is a special case of polynomial interior point cutting plane methods applied to the SDP, and analytic center cutting plane methods (ACCPM) which incorporate semidefinite cuts due to Oskoorouchi (2002) and Oskoorouchi and Goffin (2002). The latter two methods fall in the first category, while the spectral bundle method falls in the second category. Also, we note that the spectral bundle method, and the analytic center cutting plane method with semidefinite cuts constitute non-polyhedral cutting plane models for the SDP.

Consider the semidefinite programming problem

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

with dual

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

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

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

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

$$\mathcal{A}(X) \; = \; \begin{bmatrix} A_1 \bullet X \\ \vdots \\ A_{pc} \bullet X \end{bmatrix} \text{ and } \mathcal{A}^T y \; = \; \sum_{i=1}^{pc} y_i A_i$$

with $A_i \in \mathcal{S}^n, i = 1, \ldots, pc$. Here pc denotes the number of primal constraints. We assume that $A_1, \ldots, A_{pc}$ are linearly independent in $\mathcal{S}^n$. $C \in \mathcal{S}^n$ is the cost matrix, $b \in I\!R^{pc}$ the RHS vector. The matrix $X \in \mathcal{S}^n$ is constrained to be 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.

ASSUMPTION 1. *Both* (SDP) *and* (SDD) *have strictly feasible points, namely the sets* $\{X \in \mathcal{S}^n : \mathcal{A}(X) = b, X \succ 0\}$ *and* $\{(y, S) \in I\!R^{pc} \times \mathcal{S}^n : \mathcal{A}^T y + S = C, S \succ 0\}$ *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.

We will also make the following assumption, which will enable us to write (SDD) as an eigenvalue optimization problem.

ASSUMPTION 2.

$$\mathcal{A}(X) \; = \; b \; \Rightarrow \; \mathit{Trace}(X) \; = \; a \tag{1}$$

for some constant $a \geq 0$.

Assumption (2) ensures the existence of the following $\hat{y}$.

THEOREM 1. *If every feasible* X *for* (SDP) *satisfies* $\mathit{Trace}(X) = a$, *then*

$$\exists \hat{y} \in I\!R^{pc} \;\; \textit{with} \;\; \mathcal{A}^T \hat{y} = I.$$

Moreover this $\hat{y}$ *satisfies* $b^T \hat{y} = a$.

Proof: Since $\mathrm{Trace}(X) = a$ is satisfied for every feasible X in (SDP), it can be expressed as a linear combination of the other primal constraints $A_i \bullet X = b_i$, $i = 1, \ldots, pc$. Letting the components of $\hat{y}$ to be the coefficients in this linear combination we get the desired result. $\square$

We can write down the Lagrangian dual to (SDP) transferring all the equality constraints into the objective function via Lagrangian multipliers y_i, $i = 1, \ldots, pc$, to give (2). Assumption 1 ensures that this problem is equivalent to (SDP).

$$\max_y \; b^T y + \; \min_{X:\mathrm{Trace}(X)=a, X \succeq 0} \; (C - \sum_{i=1}^{pc} y_i A_i) \bullet X \qquad (2)$$

Using the variational characterization of the minimum eigenvalue function, the quantity in the inner minimization can be expressed as $a\lambda_{min}(C - \mathcal{A}^T y)$. Thus, we have

$$\max_y b^T y + a\lambda_{min}(C - \mathcal{A}^T y) \qquad (3)$$

Thus (SDP) is essentially an eigenvalue optimization problem. We shall return to (3) when we discuss cutting plane approaches for the SDP. Without loss of generality, and for the ease of exposition, we shall assume that $a = 1$ in the succeeding sections. We must also emphasize that although we are dealing with $\lambda_{min}(S)$ which is a concave function, we shall continue to use terms like subgradients, subdifferential etc. These terms should be understood to mean the corresponding analogues for a concave function. We also fix some notation here

$$\begin{array}{rcl} f(y) & = & b^T y + \lambda_{min}(C - \mathcal{A}^T y) \\ & = & \lambda_{min}(C - \sum_{i=1}^{pc} y_i (A_i - b_i I)) \end{array}$$

This function is nondifferentiable, precisely at those points, where $\lambda_{min}(C - \mathcal{A}^T y)$ has a multiplicity greater than one.

Let us consider a point y, where $\lambda_{min}(C - \mathcal{A}^T y)$ has a multiplicity r. Let p_i, $i = 1, \ldots, r$ be a normalized set of eigenvectors at this point. Also, let $P \in \mathbb{R}^{n \times r}$ be an orthonormal matrix, whose ith column is p_i. Any normalized eigenvector p corresponding to $\lambda_{min}(C - \mathcal{A}^T y)$ can be expressed as $p = Px$, where $x \in \mathbb{R}^r$, with $x^T x = 1$. The subdifferential of $f(y)$ at this point is then given by

$$\begin{array}{rcl} \partial f(y) & = & \mathrm{conv}\{b - \mathcal{A}(pp^T) : p = Px, x^T x = 1\} \\ & = & \{b - \mathcal{A}(PVP^T) : V \in \mathcal{S}^r, \mathrm{Trace}(V) = 1, V \succeq 0\} \end{array} \qquad (4)$$

Here conv denotes the convex hull operation. The equivalence of the two expressions can be found in Overton (1992). Each member of $\partial f(y)$ is called a subgradient.

We also define the ϵ subdifferential at a point y, which we denote as $\partial_\epsilon f(y)$. This is given by

$$\begin{aligned}\partial_\epsilon f(y) \;=\; &\{b - \mathcal{A}(pp^T) : (b - \mathcal{A}(pp^T))^T(x-y) \\ &\geq \lambda_{min}(S(x)) - \lambda_{min}(S(y)) - \epsilon, \forall S(x) \in \mathcal{S}^n\}\end{aligned} \tag{5}$$

2. Linear programming formulations of semidefinite programming

In this section we shall discuss linear programming cutting plane schemes for the SDP. We shall motivate these cutting plane schemes from two perspectives. We analyze the shortcomings of these cutting plane schemes, and then consider modifications, which lead to an improved polyhedral bundle scheme for the SDP. Further enhancements are possible, based on the multiplicity of $\lambda_{min}(S)$, and these lead to non-polyhedral cutting plane models for the SDP. We shall discuss two of these approaches in the next section.

The first way to motivate cutting plane linear programming approaches for the (SDP) is to consider a *semi-infinite* formulation (LDD) of (SDD). This formulation is given below.

$$\begin{array}{lrcll} \max & b^Ty & & & \\ \text{subject to} & \mathcal{A}^Ty + S & = & C & \qquad (LDD) \\ & d^TSd & \geq & 0 & \forall ||d||_2 = 1 \end{array}$$

We consider (SDD) instead of (SDP) for the following reasons:

- Since X is a $n \times n$, and symmetric matrix, a semi-infinite formulation of (SDP) would involve $\binom{n+1}{2} = \frac{n(n+1)}{2} = O(n^2)$ variables.
- There are pc variables in the dual formulation, and we have $pc \leq \binom{n+1}{2}$. Thus it is more efficient to deal with the dual semi-infinite formulation, since we are dealing with smaller linear programs.

We discuss the finite linear programs (LDR) and (LPR) and some of their properties below. Given a finite set of vectors $\{d_i, i = 1, \ldots, m\}$, we obtain the relaxation

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

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

$$d_id_i^T \bullet \mathcal{A}^Ty \;=\; d_id_i^T \bullet (\sum_{j=1}^{pc} y_jA_j)$$

$$= \sum_{j=1}^{pc} y_j d_i^T A_j d_i.$$

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

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

It follows that the dual problem is

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

This can be rewritten as

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

THEOREM 2. *Any feasible solution x to (LPR) will give a feasible solution X to (SDP).*

Proof: This lemma follows directly from the fact that (LPR) is a constrained version of (SDP). However we present a formal 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

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

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

We discuss the perfect set of linear constraints that are required for the SDP.

The optimality conditions for the SDP include primal feasibility, dual feasibility and complementarity $XS = 0$. The complementarity condition implies that X and S commute, and so they share a common share of eigenvectors (see *simultaneous diagonalization* in Horn and Johnson (1990)).

Thus, we have (Alizadeh et al., 1997):

THEOREM 3. *Let X and (y, S) be primal and dual feasible respectively. Then they are optimal if and only if there exists $Q \in \mathbb{R}^{n \times r}$, $R \in \mathbb{R}^{n \times (n-r)}$, with $Q^T Q = I_r$, $R^T R = I_{n-r}$, $Q^T R = 0$, and Λ, Ω, diagonal matrices in ${\mathcal{S}_+}^r$, and ${\mathcal{S}_+}^{n-r}$, such that*

$$X = [Q \ \ R] \begin{bmatrix} \Lambda & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} Q^T \\ R^T \end{bmatrix} \tag{6}$$

$$S = [Q \;\; R] \begin{bmatrix} 0 & 0 \\ 0 & \Omega \end{bmatrix} \begin{bmatrix} Q^T \\ R^T \end{bmatrix} \tag{7}$$

hold.

The diagonal matrices Λ, Ω contain the nonzero eigenvalues of X and S in the spectral decompositions (6) and (7) respectively. Also $P = [Q \;\; R]$ is an orthogonal matrix that contains the common set of eigenvectors. Note that we are assuming that strict complementarity holds here ((Alizadeh et al., 1997)).

We get an upper bound on r, using the following theorem 4, due to Pataki (1998) (also see Alizadeh et al. (1997)), on the rank of extreme matrices X in (SDP).

THEOREM 4. *There exists an optimal solution X^* with rank r satisfying the inequality $\frac{r(r+1)}{2} \leq pc$, where pc is the number of constraints in (SDP).*

Theorem 4 suggests that there is an optimal matrix X that satisfies the upper bound, whose rank is around $O(\sqrt{pc})$.

THEOREM 5. *Let $X^* = Q\Lambda Q^T$, and let q_i, $i = 1, \ldots, r$ be the columns of Q. Then the optimal solution to (SDD) and (SDD) is given by the optimal solution to (8), and its dual (9).*

$$\begin{array}{lrcl} \max & & b^T y & \\ s.t. & \sum_{j=1}^{pc}(q_i^T A_j q_i) y_j & \leq & q_i^T C q_i, \;\; i = 1, \ldots, r \end{array} \tag{8}$$

with dual

$$\begin{array}{lrcl} \min & \sum_{i=1}^{r}(q_i^T C q_i) x_i & & \\ s.t. & \sum_{i=1}^{r}(q_i^T A_j q_i) x_i & = & b_j \;\; j = 1, \ldots, pc \\ & x & \geq & 0 \end{array} \tag{9}$$

Proof: Since (8) is a discretization to (SDD), its dual (9) is a constrained version of (SDP). Thus its optimal value gives an upper bound on the optimal value of (SDP). Thus an optimal solution to (SDP) is also optimal in (8), provided it is feasible in it. The optimal solution to (SDP) is given by $X = Q\Lambda Q^T = \sum_{i=1}^{r} \lambda_i q_i q_i^T$, where $\lambda_i > 0$, $i = 1, \ldots, r$, and q_i, $i = 1, \ldots, r$ are the corresponding eigenvectors. This is clearly feasible in (9), with $x = \lambda$. This corresponds to (LDR) with $d_i = q_i$, $i = 1, \ldots, r$. $\square$

Theorem 5 tells us precisely what constraints we should look for in our LP relaxations, namely those in the *null space of the optimal dual slack matrix S.*

We conclude this section with noting that (9) can be rewritten as

$$\begin{array}{lrcl} \min & C \bullet (QMQ^T) & & \\ \text{s.t.} & A_j \bullet (QMQ^T) & = & b_j \;\; j = 1, \ldots, pc \\ & M & \succeq & 0 \\ & M & \text{diagonal} & \end{array} \tag{10}$$

Here $M \in \mathcal{S}^m$. Choosing $m = r$ gives the formulation in (9). Although the constraint set in (9) is an overdetermined system, we still have a solution.

In the LP approach, the number of columns of columns in Q can get arbitrarily large. However when this number is more than n, we could instead perform a spectral factorization of $X = QMQ^T = P\Lambda P^T$, and replace the columns of Q with the set of eigenvectors in P.

3. Nonpolyhedral cutting plane models for semidefinite programming

We discuss extensions to the polyhedral model presented in section 2. These are primarily motivated by (10). A natural relaxation of (10) is to drop the requirement that M be diagonal, giving rise to (11).

$$\begin{array}{lrcl} \min & C \bullet (QMQ^T) & & \\ \text{s.t.} & A_j \bullet (QMQ^T) & = & b_j \;\; j = 1, \ldots, pc \\ & M & \succeq & 0 \end{array} \tag{11}$$

with dual

$$\begin{array}{lrcl} \min & b^T y & & \\ \text{s.t.} & S & = & Q^T(C - \mathcal{A}^T y)Q \\ & S & \succeq & 0 \end{array} \tag{12}$$

where $M \in \mathcal{S}^m$. Incidentally, this is a constrained version of (SDP) for $m \leq n$. However, (11) is equivalent to (SDP), when the columns of Q are p_i from Theorem (5). Typically, we would like $m = r = O(\sqrt{pc})$, where pc is the number of constraints in (SDP), since we interested only in the set of optimal solutions.

It appears from (11) that we are solving an SDP by solving a sequence of smaller SDP's. We would like to think of solving (11) as a *primal active set* method for the SDP.

1. Consider a subset $X = QMQ^T$ of the primal feasible matrices, with $M \in \mathcal{S}^{pc}$. We are exploiting Theorem 4 on the rank of basic feasible X matrices here.

2. This amounts to relaxing $S \succeq 0$ to $Q^T(C - \mathcal{A}^T y)Q \succeq 0$ in (12). Compute the most negative eigenvalue $\lambda_{min}(S)$ together with an associated eigenvector v. If S is psd, we stop with optimality.

3. Update Q to include the new vector v, and using some strategy to keep the number of columns in Q constant. This corresponds to another basic feasible solution, with hopefully with a better objective value. We return to step 1.

We shall retun to (11) when we consider the spectral bundle method (Algorithm 4), and a primal active set method for SDP (Algorithm 5) in section 5.

There is another formulation (13), which is in between (10) and (11).

$$\begin{array}{lrcl} \min & C \bullet (QMQ^T) & & \\ \text{s.t.} & A_j \bullet (QMQ^T) & = & b_j \;\; j = 1, \ldots, pc \\ & M & \succeq & 0 \\ & M & \multicolumn{2}{l}{\text{block diagonal}} \end{array} \tag{13}$$

We will later relate (13) with the analytic center cutting plane method incorporating semidefinite cuts due to Oskoorouchi and Goffin (2002) and Oskoorouchi (2002).

4. Cutting plane algorithms for semidefinite programming

We now present a preliminary LP cutting plane algorithm for the SDP based on a semi-infinite formulation of (LDD). The method is originally due to Cheney and Goldstein (1959), and Kelley (1960). This is also the exchange scheme discussed in Hettich and Kortanek (1993). We shall henceforth refer to this as Algorithm 1 and it is presented below.

We wish to emphasize a few points with regard to Algorithm 1.

- An initial set of constraints is obtained by requiring that the diagonal entries of S be non-negative. This amounts to setting $d = e_i$, $i = 1, \ldots, n$.

- We solve subproblem (14) cheaply. Typically, the most negative eigenvalue $\lambda_{min}(S)$, and its associated eigenvector is estimated by an iterative method like the *Lanczos* scheme.

- Any d which gives a negative objective value in (14) gives a valid cutting plane. In particular, this includes all eigenvectors corresponding to negative eigenvalues.

1. Choose an **initial set** of constraints for (LDR). Choose termination parameters $\epsilon_1, \epsilon_2 > 0$. Set the current upper and lower bounds to be UB $= \infty$, and LB $= -\infty$ respectively.

2. In the kth iteration we solve a discretization (LDR), and its dual (LPR) for a solution $\bar{y}^k$ where

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

   with dual

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

   Update the upper bound: UB $= \min\{\text{UB}, b^T \bar{y}^k\}$.

3. Solve the subproblem

$$\begin{array}{ll} \min & d^T(C - \mathcal{A}^T \bar{y}^k) d \\ \text{s.t.} & \qquad ||d||_2 \ \leq \ 1 \end{array} \qquad (14)$$

   The optimal objective value to (14) is $\lambda_{min}(C - \mathcal{A}^T \bar{y}^k)$. Update the lower bound: LB $= \max\{\text{LB}, b^T(\bar{y}^k + \lambda \hat{y})\}$, where $\lambda = |\lambda_{min}(C - \mathcal{A}^T \bar{y}^k)|$, and $\hat{y}$ satisfies Theorem 1. If $|\text{LB} - \text{UB}| \leq \epsilon_1$, or $\lambda \leq \epsilon_2$, go to step 5.

4. Any eigenvector d corresponding to this eigenvalue $\lambda_{min}(C - \mathcal{A}^T \bar{y}^k)$ gives a valid cutting plane.

$$dd^T \bullet \mathcal{A}^T y \ \leq \ dd^T \bullet C$$

   Add this cutting plane to (LDR). Set $k = k + 1$, and return to step 2.

5. The current solution (x^k, y^k) gives an optimal solution (X, y) for (SDP), and (SDD) respectively, where $X = \sum_{i=1}^{m^k} {x_i}^k d_i d_i^T$, and $y = y^k$.

Figure 1. Algorithm 1: Cutting plane algorithm for the SDP

– We could choose a different norm for d in (14). The reason for choosing the 2 norm is that (14) can be solved efficiently, thanks to the variational characterization of the minimum eigenvalue function.

There is another way to motivate this cutting plane approach, which is based on the eigenvalue optimization model (3).

Now assume that we have a set of points $y = y^1, \ldots, y^m$, and we know the function values $f(y^i)$, $i = 1, \ldots, m$, and subgradients $(b - \mathcal{A}(p_i p_i^T))$, $i = 1, \ldots, m$ (where p_i is a normalized eigenvector corresponding to $\lambda_{min}(C - \mathcal{A}^T y^i)$) at these points.

We can construct the following overestimate $\hat{f}_m(y)$ for $f(y)$.

$$\begin{array}{rcl} \hat{f}_m(y) & = & \min_{i=1,\ldots,m} p_i p_i^T \bullet (C - \mathcal{A}^T y) + b^T y \\ & \geq & f(y) \end{array}$$

To see this note that since the p_i are normalized, we have

$$\begin{array}{rcl} \lambda_{min}(C - \mathcal{A}^T y) & \leq & p_i^T (C - \mathcal{A}^T y) p_i, \quad i = 1, \ldots, m \\ & = & p_i p_i^T \bullet (C - \mathcal{A}^T y), \quad i = 1, \ldots, m \end{array}$$

We now maximize this overestimate instead, i.e.

$$\max_y \hat{f}_m(y) \; = \; \max_y \{ b^T y + \min_{i=1,\ldots,m} \{ p_i p_i^T \bullet (C - \mathcal{A}^T y) \} \}$$

which can be recast as the following linear program

$$\begin{array}{ll} \max & b^T y + v \\ \text{s.t.} & v \leq p_i p_i^T \bullet (C - \mathcal{A}^T y), \quad i = 1, \ldots, m \end{array} \tag{15}$$

with dual

$$\begin{array}{lrcl} \min & \sum_{i=1}^m (p_i^T C p_i) x_i & & \\ \text{s.t.} & \sum_{i=1}^m (p_i^T A_j p_i) x_i & = & b_j, \; j = 1, \ldots, pc \\ & \sum_{i=1}^m x_i & = & 1 \\ & x & \geq & 0 \end{array} \tag{16}$$

This is exactly the problem obtained by considering a discretization of (SDD). Here v is the dual variable corresponding to the constraint $\sum_{i=1}^m x_i = 1$, which is implicitly satisfied by any solution x to (LPR) (since we assumed that any feasible X in (SDP) satisfies $\mathrm{Trace}(X) = 1$, and (LPR) is a constrained version of (SDP)). Thus, we can set $v = 0$ without any loss of generality. In fact we must mention that (15) has an infinite number of solutions, since we added the redundant constraint $\sum_{i=1}^m x_i = 1$ in (16) corresponding to $\mathrm{Trace}(X) = 1$ in Assumption 2. The solution (v, y) with $v = 0$ is the one corresponding to (LDR).

Unfortunately Algorithm 1 has a very poor rate of convergence. For instance we observed that a simplex implementation performed very badly (Krishnan and Mitchell, 2001; Krishnan, 2002). Primarily, minimizing $\hat{f}_m$ to find y^{m+1} makes sense only if $\hat{f}_m \approx f$, near y^m, this is one of the reasons for the slow convergence for the cutting plane scheme. Lemarechal (1989) discusses some convergence estimates for such an algorithm. Secondly the number of constraints in Algorithm 1 gets prohibitively large. One way to overcome this is to utilize the refactorization idea we mentioned at the end of section 2.

There are two ways of addressing the shortcoming of slow convergence:

– Utilize an interior point cutting plane scheme that solves (LDR) and (LPR) approximately, i.e. to some tolerance on the duality gap $\frac{x^T s}{n}$. As the algorithm proceeds, we gradually tighten this tolerance. This is the interior point LP cutting plane scheme discussed in Krishnan and Mitchell (2001). (See also Mitchell (2000), Mitchell and Ramaswamy (2000), and Goffin and Vial (2002) for more discussions on interior point cutting plane algorithms). It is hard to prescribe explicit rules for updating this tolerance. A rough rule of thumb would be lower the tolerance if the cutting plane algorithm seems to be doing well, and actually increase this if the algorithm seems to be hopelessly stuck somewhere. We wish to relate these two stages with the serious and null steps carried out in the proximal bundle scheme discussed below.

– The second idea is to utilize the proximal bundle idea as discussed in Lemarechal (1989) and Kiwiel (1985). We present a short discussion on the proximal bundle scheme below. The rough idea here is to maximize $\hat{f}_m(y) - \frac{u}{2}||y - y^m||^2$ (for some chosen $u > 0$). The second term acts as a regularization term which penalizes us from going too far from the current iterate y^m. The idea is to lower u if we are making progress, i.e. taking serious steps, and actually increase u if we perform a null step. As Lemarechal (1989) remarks, choosing this parameter u is an art in itself. The regularization penalty term $\frac{u}{2}||y - y^m||^2$ acts as a trust region constraint $||y - y^m||^2 \leq \sigma_m$, and helps to keep the solution bounded. Thus we can dispense with choosing an initial set of constraints to keep the subproblems bounded, as in Algorithm 1. For numerical reasons, it is better to introduce the regularization term into the objective function, rather than as a trust region constraint. This keeps the feasible region polyhedral, but we now have a quadratic objective. Also, this entails the choice of a good $u > 0$.

We now present a discussion of the proximal bundle scheme. Some excellent references include Hiriart-Urruty and Lemarechal (1993), the monograph by Kiwiel (1985), and the survey article by Lemarechal (1989). We begin from where we left off from the cutting plane scheme. To simplify our notation, we assume that y^m is the iterate where $f(y)$ is a maximum, i.e. $f(y^m) \geq f(y^i)$, $i = 1, \ldots, m-1$. Let us also introduce some notation here. Let

$$\begin{aligned} q_i &= b^T(y^m - y^i) + (\lambda_{min}(C - \mathcal{A}^T y^m) - \lambda_{min}(C - \mathcal{A}^T y^i)) \\ &\qquad -(b - \mathcal{A}(p_i p_i^T))^T(y^m - y^i) \\ d &= y - y^m \end{aligned}$$

Here q_i refers to the linearization error due to the subgradient $(b - \mathcal{A}(p_i p_i^T))$ (computed at y^i), at the current iterate y^m. Since we are dealing with a concave function, $q_i \leq 0$, $i = 1, \ldots, m-1$. Using the above notation, we can write $\hat{f}_m(y)$ as follows:

$$\hat{f}_m(y) = f(y^m) + \min\{-q_i + (b - \mathcal{A}(p_i p_i^T))^T d, i = 1, \ldots, m\} \qquad (17)$$

with $q_m = 0$. The final problem we solve to obtain the search direction d is then

$$\begin{array}{ll} \max & v - \frac{1}{2} u d^T d \\ \text{s.t.} & v \leq -q_i + (b - \mathcal{A}(p_i p_i^T))^T d, \quad i = 1, \ldots, m \end{array} \qquad (18)$$

If we set $u = 0$ in (18) we identically have (15). The dual to this subproblem is (19). Due to strong duality the two problems (18) and (19) are identical.

$$\begin{array}{ll} \min & \frac{1}{2u} ||b - \mathcal{A}(\sum_{i=1}^m x_i p_i p_i^T)||^2 - \sum_{i=1}^m x_i q_i \\ \text{s.t.} & \sum_{i=1}^m x_i = 1 \\ & x \geq 0 \end{array} \qquad (19)$$

The first term is obtained by noting that $(b - \mathcal{A}(\sum_{i=1}^m x_i p_i p_i^T))^T d - \frac{u}{2} d^T d$ is a strictly concave function and attains its maximum when

$$d = \frac{1}{u}(b - \mathcal{A}(\sum_{i=1}^{m-1} x_i p_i p_i^T)) \qquad (20)$$

We describe the polyhedral bundle method for the SDP below, in Algorithm 2.

A few points are now in order: We are considering a subset of the positive semidefinite matrices X in (SDP) with trace one. To see this note that $X = \sum_{i=1}^m x_i p_i p_i^T$ is positive semidefinite, with trace one since $\sum_{i=1}^m x_i = 1$. When the method converges to the optimal solution, i.e. $d = 0$, we get primal feasibility, since in addition to $X \succeq 0$ we have $\mathcal{A}(X) = b$.

1. Start with $y^1 \in I\!R^{pc}$, let $p^1 \in I\!R^n$ be a normalized eigenvector corresponding to $\lambda_{min}(C - \mathcal{A}^T y^1)$. Also choose the weight $u > 0$, an improvement parameter ν_1 satisfying $0 < \nu_1 < 1$, and finally a termination parameter $\epsilon > 0$.

2. In the k iteration, compute the search direction d^k using (20), where x solves (19).

3. If $||d^k|| < \epsilon$, then stop.

4. Perform a line search along the direction d^k, and set $\bar{y}^{k+1} = y^k + td^k$. Compute p^{k+1}, a normalized eigenvector corresponding to $\lambda_{min}(C - \mathcal{A}^T \bar{y}^{k+1})$.

5. If the actual increase is not much smaller than the increase predicted by the model (sufficient increase), i.e.

$$f(\bar{y}^{k+1}) - f(y^k) \; \geq \; \nu_1 t(\hat{f}_k(\bar{y}^{k+1}) - f(y^k))$$

   then perform a serious step, i.e. $y^{k+1} = \bar{y}^{k+1}$. Here $\hat{f}_k(\bar{y}^{k+1})$ is the objective value of (19). Set $q_{k+1} = 0$, and adjust q_i, $i = 1, \ldots, k$ as follows

$$q_i \; = \; q_i + f(y^{k+1}) - f(y^k) - t(b - \mathcal{A}(p_i p_i^T))^T d^k$$

6. Else, perform a null step, i.e. $y^{k+1} = y^k$.

7. Return to step (2).

Figure 2. Algorithm 2: Polyhedral bundle method for SDP

We refer to the collection of subgradients $(b - \mathcal{A}(p_i {p_i}^T))$ as the bundle. The size of this bundle r is the number of these subgradients. It appears that this size grows indefinitely with iteration count k in the above algorithm. We can however keep the bundle size bounded, by introducing an aggregate subgradient matrix $\bar{W}$. We illustrate one possibility for this for the $(k+1)$ iteration in the above algorithm as follows:

1. Aggregate all the information from the previous iterates in an *aggregate subgradient* $\bar{W}$ as follows:

$$\bar{W}^k \; = \; \textstyle\sum_{i=1}^m x_i p_i p_i^T$$

2. Thus in the $(k+1)$th iteration, the bundle consists of only two elements $(b - \mathcal{A}(\bar{W}^k))$, and $(b - \mathcal{A}(p^{k+1} {p^{k+1}}^T))$.

3. For the aggregation considered above, we are considering a subset of the positive semidefinite matrices X with trace one. This subset $\mathcal{W}^{k+1}$ is given by

$$\mathcal{W}^{k+1} \;=\; \mathrm{conv}\{\bar{W}^k, p^{k+1}{p^{k+1}}^T\} \tag{21}$$

If we define $\epsilon_k = -\sum_{i=1}^m x_i q_i$ (which appears in the objective function), then this implies the subgradients $(b - \mathcal{A}(p_i p_i^T))$, $i = 1, \ldots, m$, corresponding to nonzero x_i are ϵ_m subgradients of the minimum eigenvalue function at the current point y^k. This tolerance controls the radius of the ball in which we consider the bundle model to be a good approximation of the objective function $f(y)$. We can think of the weight parameter u to be the dual multiplier associated with the constraint $-\sum_{i=1}^m x_i q_i \le \epsilon_k$. The main difficulty in the bundle method is in choosing the weight $u > 0$, or equivalently fixing the tolerance ϵ_k.

The polyhedral bundle scheme constructs the following polyhedral approximation $\mathcal{P}^k$ of $\partial f(y^k)$ in the kth iteration.

$$\begin{aligned}\mathcal{P}^k \;=\; \{& b - \mathcal{A}(pp^T) : p = \textstyle\sum_{i=1}^m x_i p_i p_i^T, -\sum_{i=1}^m x_i q_i \le \epsilon_k, \\ & \textstyle\sum_{i=1}^m x_i = 1, x \ge 0\}\end{aligned} \tag{22}$$

This follows from setting $q_i = 0, \forall i$ in (22). To be technically more precise we must say that $\mathcal{P}^k$ attempts to be a good approximation of $\partial_\epsilon f(y^k)$. We omit these details, and they can be found in Lemarechal (1989) and Makela and Neittaanmaki (1992). In fact, this is needed for the actual convergence of the algorithm. Thus we are able to construct a good approximation to $\partial_{\epsilon_m} f(y^k)$ despite using the knowledge of only one subgradient in every iteration.

We will now replace (19) with the following problem (23) in the kth iteration using the explanation below, where

$$\begin{array}{lrcl} \min & \frac{1}{2}\|b - \mathcal{A}(X)\|^2 & & \\ \text{s.t.} & X & = & \sum_{i=1}^m x_i p_i p_i^T \\ & \sum_{i=1}^m x_i & = & 1 \\ & x & \ge & 0 \end{array} \tag{23}$$

This can be interpreted as utilizing the subgradient $(b - \mathcal{A}(pp^T))$ in computing the search direction d, where pp^T is given the convex combination of the $p_i p_i^T$, and pretending that they belong to $\partial f(y^k)$, i.e. $q_i = 0$. In a practical implementation, this is saying that we choose the p_i, whose q_i are quite small. We shall hereafter refer to (23) as the subproblem handled by the bundle method in each iteration. In any case, this is handled via an appropriate choice of the weight parameter $u > 0$.

5. Non-polyhedral cutting plane models for the SDP

In this section, we discuss non-polyhedral cutting plane models for the SDP. We discuss two of them:

- The analytic center cutting plane method, which employs semidefinite cuts (Oskoorouchi and Goffin, 2002; Oskoorouchi, 2002). This method is an extension of Algorithm 1; however the primal subproblem solved in every iteration is no longer an LP, but an SDP with a block diagonal structure. We discuss this scheme in section 5.1.
- The spectral bundle method (Helmberg and Rendl, 2000; Helmberg, 2000; Helmberg and Kiwiel, 1999; Helmberg and Oustry, 2000; Oustry, 2000).This method is an extension of Algorithm 2, which utilizes the non-polyhedral expression for the subdifferential of $\lambda_{min}(S)$. We discuss this in section 5.2.

5.1. A nonpolyhedral cutting plane algorithm for SDP

We begin our discussion with the first scheme. More details can be found in Oskoorouchi and Goffin (2002) and Oskoorouchi (2002). We shall henceforth refer to the scheme as Algorithm 3, shown below. It is a natural extension of Algorithm 1.

Subproblem (25) involves solving a conic optimization problem over the intersection of linear and SDP blocks. Each SDP block is no bigger than $O(\sqrt{pc})$. This is the formulation (13) we discussed in section 3. It is imperative to keep the size of X small, so that this SDP can be solved quickly.

5.2. The spectral bundle method for SDP

The spectral bundle method is based on the same ideas as the proximal bundle method discussed in the previous section, the only difference being the bundle method operates with the second expression in (4) for the subdifferential of the minimum eigenvalue function. Note that this set is no longer polyhedral, but is actually an ellipsoid. Thus the spectral bundle method constitutes a non-polyhedral cutting plane scheme for solving the SDP. We present a cursory version of the spectral bundle method below (Algorithm 4). For more details we refer the reader to Helmberg and Rendl (2000), Helmberg (2000), Helmberg and Kiwiel (1999), Helmberg and Oustry (2000), and Oustry (2000).

The subproblem (26) the bundle solves in every iteration bears a close similarity to (11) we discussed in section 3. (without the quadratic

1. Choose an **initial set** of constraints for (LDR), i.e.

$$\sum_{i=1}^{pc} y_i(d_j^T A_i d_j) \ \leq \ d_j^T C d_j, \ \ j = 1, \ldots, m$$

   Set $D = [d_1, \ldots, d_m]$.

2. In the kth iteration solve the following pair of SDP's for (X^k, y^k).

$$\begin{array}{lrcl} \max & & b^T y & \\ \text{s.t.} & \sum_{i=1}^{pc} y_i(d_j^T A_i d_j) & \leq & d_j^T C d_j, \quad j = 1, \ldots, m \\ & \sum_{i=1}^{pc} y_i(D_j^T A_i D_j) & \preceq & D_j^T C D_j, \ \ j = 1, \ldots, m^k \end{array} \tag{24}$$

   with dual

$$\begin{array}{lrcl} \min & C \bullet (DXD^T) & & \\ \text{s.t.} & A_i \bullet (DXD^T) & = & b_i, \ \ i = 1, \ldots, pc \\ & x_i & \geq & 0, \ \ i = 1, \ldots, m \\ & X_i & \succeq & 0, \ \ i = 1, \ldots, k \end{array} \tag{25}$$

   where

$$X \ = \begin{bmatrix} x_1 & & & & & \\ & \ddots & & & & \\ & & x_m & & & \\ & & & X_1 & & \\ & & & & \ddots & \\ & & & & & X_k \end{bmatrix}$$

3. Compute $\lambda_{min}(C - \mathcal{A}^T y^k)$, and an associated set of eigenvectors d_i^k, $i = 1, \ldots, r^k$. Here r^k is the multiplicity of this eigenvalue. Typically $r^k \leq O(\sqrt{pc})$. Also, update the lower and upper bounds as in Algorithm 1. If $\lambda_{min}(C - \mathcal{A}^T y^k)$ is small, or the difference between the computed bounds is less than the specified tolerance go to step 4. Else, set $D_k = [d_1^k, \ldots, d_{r^k}^k]$. This gives the valid cutting plane

$$\sum_{i=1}^{pc} y_i(D_k^T A_i D_k) \ \preceq \ D_k^T C D_k$$

   Add this SDP constraint to (24), and update $D = [D; D_k]$. Set $k = k + 1$, and return to step 2.

4. The current solution (X^k, y^k) is optimal for (SDP), and (SDD) with $X^* = DX^k D^T$, and $y^* = y^k$ respectively.

Figure 3. Algorithm 3: Non-polyhedral cutting plane algorithm for SDP

1. Start with $y^1 \in I\!R^k$, let $p^1 \in I\!R^n$ be a normalized eigenvector corresponding to $\lambda_{min}(C - \mathcal{A}^T y^1)$. Also choose the weight $u > 0$, an improvement parameter ν_1 satisfying $0 < \nu_1 < 1$, and finally a termination parameter $\epsilon > 0$. Also, let $P^1 = p^1$, and $\bar{W}^1 = p^1 {p^1}^T$.

2. In the kth iteration, solve the following subproblem (26) for (α^k, V^k, X^k).

$$\begin{array}{lrcl} \min & \frac{1}{2}\|b - \mathcal{A}(X)\|^2 & & \\ \text{s.t.} & X & = & \alpha \bar{W}^k + P^k V (P^k)^T \\ & \alpha + \mathrm{Trace}(V) & = & 1 \\ & V & \succeq & 0 \\ & \alpha & \geq & 0 \end{array} \tag{26}$$

   Compute the search direction $d^k = \frac{1}{u}(b - \mathcal{A}(X^k))$.

3. If $\|d^k\| < \epsilon$, then stop.

4. Compute $\bar{y}^{k+1} = y^k + d^k$, $\lambda_{min}(C - \mathcal{A}^T \bar{y}^{k+1})$, and an associated eigenvector p^{k+1}.

5. If the actual increase is not much smaller than the increase predicted by the model, i.e.

$$f(\bar{y}^{k+1}) - f(y^k) \;\geq\; \nu_1 t(\hat{f}_k(\bar{y}^{k+1}) - f(y^k))$$

   then perform a serious step, i.e. $y^{k+1} = \bar{y}^{k+1}$. Here $\hat{f}_k(\bar{y}^{k+1})$ is the objective value of (26).

6. Else, set $y^{k+1} = y^k$.

7. Compute $V^k = Q \Lambda Q^T$. Split $Q = [Q_1, Q_2]$, where Q_1, and Q_2 contain the eigenvectors corresponding to the large (Λ_1), and small (Λ_2) eigenvalues of V^k respectively. Finally update the bundle P, and the aggregate matrix $\bar{W}$ as follows:

$$\begin{array}{rcl} P^{k+1} & = & \mathrm{orth}([P^k Q_1, v^{k+1}]) \\ \bar{W}^{k+1} & = & \frac{1}{\alpha^k + \mathrm{Trace}(\Lambda_2)}(\alpha^k \bar{W}^k + P^k Q_2 \Lambda_2 (P^k Q_2)^T) \end{array} \tag{27}$$

8. Set $k = k + 1$, and return to step 2.

Figure 4. Algorithm 4: The spectral bundle method for SDP

regularization term). The only difference between the spectral bundle, and the polyhedral bundle method is in the bundle subproblem that is solved in each iteration: for the polyhedral bundle this is (23), for the spectral bundle method this is (26) an SDP with a quadratic objective function. When the number of columns in P becomes large, then solving (26) is almost as difficult as solving the original SDP. To overcome this difficulty, the less important subgradient information is aggregated in $\bar{W}$ which plays the role of an aggregate subgradient. This helps to keep the number of columns r in P small, and still guarantee the convergence of the algorithm. The matrix P is the bundle here, and contains the more important subgradients $(b - \mathcal{A}(p_i p_i^T))$, where p_i, $i = 1, \ldots, r$ are the columns of P.

On comparing (26) with (23) we note the following:

- In (23) we have to solve a quadratic programming problem over a polyhedral region. In (26) we have an SDP with a quadratic objective function.

- For the extreme case in the spectral bundle method, where we maintain only the latest subgradient in the bundle P, and aggregate all the previous information in the matrix $\bar{W}$, the two methods are exactly the same. In the $(k+1)$th iteration this corresponds to choosing an X from the set $\mathcal{W}^k$, given in (21).

- The spectral bundle method affords other means of aggregation, in order to keep the number of columns in P small. This is described in step 7 of Algorithm 4. Note that $X = PVP^T$ is not a spectral decomposition of X, since V is not diagonal. The actual decomposition is $X = (PQ)\Lambda(PQ)^T$, where $V = Q\Lambda Q^T$. The aggregation is crucial to the spectral bundle scheme, since we have to solve a quadratic SDP in each iteration.

- Let $\mathcal{W}_{sb}^k$ denote the feasible region of the spectral bundle subproblem (26), and $\mathcal{W}_{pb}^k$ the feasible region of the polyhedral bundle subproblem (23) in the kth iteration of each method. We always have
$$\mathcal{W}_{pb}^k \subseteq \mathcal{W}_{sb}^k$$
This can be easily explained by noting that we need an infinite number of terms in $\mathcal{W}_{pb}^k$ to capture the smoothness inherent in $\mathcal{W}_{sb}^k$.

We close this paper with a final non-polyhedral cutting plane algorithm for the SDP, Algorithm 5 below.

1. Choose $\bar{W}^1 \in \mathcal{S}^n$ to be any feasible matrix in (SDP), and any $y^1 \in I\!R^{pc}$.

2. In the kth iteration solve (28), and its dual (29) for $(\alpha^k, V^k, y^k) \in I\!R \times \mathcal{S}^r \times I\!R^{pc}$.

$$\begin{array}{lrcl} \min & C \bullet (\alpha \bar{W}^k + P^k V P^{k^T}) & & \\ \text{s.t.} & A_i \bullet (\alpha \bar{W}^k + P^k V P^{k^T}) & = & b_i, \ i = 1, \ldots, pc \\ & \alpha & \geq & 0 \\ & V & \succeq & 0 \end{array} \tag{28}$$

with dual

$$\begin{array}{lrcl} \max & & b^T y & \\ \text{s.t.} & \sum_{i=1}^{pc} y_i (P^{k^T} A_i P^k) & \preceq & P^{k^T} C P^k \\ & \sum_{i=1}^{pc} y_i (A_i \bullet \bar{W}^k) & \leq & C \bullet \bar{W}^k \end{array} \tag{29}$$

3. Compute $\lambda_{min}(C - \mathcal{A}^T y^k)$, and its associated eigenvector v^k. Update the lower and upper bounds as discussed in Algorithm 1. If $\lambda_{min}(C - \mathcal{A}^T y^k)$ is small, or the difference between the upper and lower bounds is less than the specified tolerance go to step 4. Else compute the spectral factorization $V^k = Q \Lambda Q^T$, with $0 \leq \lambda_1 \leq \ldots \leq \lambda_r$. Let q_1 denote the first column of Q and let $Q(2:r)$ denote the remaining columns of Q, so $V^k = \lambda_1 q_1 q_1^T + Q(2:r)\Lambda(2:r)Q(2:r)^T$, with $\Lambda(2:r)$ defined appropriately. Set:

$$\begin{array}{rcl} P^{k+1} & = & \mathrm{orth}([P^k Q(2:r), \ v]) \\ \bar{W}^{k+1} & = & \frac{1}{(\alpha^k + \lambda_1)} (\alpha^k \bar{W}^k + \lambda_1 P^k q_1 q_1 P^{k^T}), \end{array}$$

where $\mathrm{orth}(M)$ denotes the elements of an orthogonal basis of the range of M. Return to Step 2.

4. The current solution (α^k, V^k, y^k) is optimal for (SDP), and (SDD) with $X^* = \alpha^k \bar{W}^k + P^{k^T} V^k P^k \approx P^{k^T} V^k P^k$, and $y^* = y^k$ respectively.

Figure 5. Algorithm 5: A primal active set method for SDP

Algorithm 5 resembles the primal active set method for LP. We are always primal feasible (unlike the spectral bundle method in Algorithm 4). Also we are solving a linear SDP, unlike a quadratic SDP in Algorithm 4. We are dual feasible only at optimality. In fact a negative eigenvalue in Step 3 is like a negative reduced cost in the primal simplex method. To speed up Algorithm 5, we add more than just one eigenvector corresponding to a negative eigenvalue in Step 3. The convergence, and computational aspects of Algorithm 5 are discussed in a forthcoming paper (Krishnan and Mitchell, 2003). We hope this will lead eventually to simplex-like methods for the SDP (see Pataki (1996) for an alternative derivation of a simplex-type method for SDP).

We close the discussion by reiterating an important advantage of non-polyhedral algorithms such as Algorithms 4 and 5 based on (11) over polyhedral Algorithms 1 and 2 which work with (10).

We terminate in the non-polyhedral algorithms when we have the right subspace P over which the dual slack matrix S is required to be psd. In fact this subspace provides a basis for the eigenvectors corresponding to the strictly positive eigenvalues of optimal primal matrix X^*. (note $X^* = PVP^T = (PQ)\Lambda(PQ)^T$).

On the other hand, the polyhedral approaches require the exact eigenvectors corresponding to the strictly positive eigenvalues of X^*, which is a more stringent requirement, and hence more difficult to achieve.

6. Conclusions

We present an accessible and unified introduction to various cutting plane methods that have appeared in the literature. We discuss five algorithms in all, of increasing complexity, that build on their predecessors. In fact, Algorithms 2, 3, 4, and 5 arise as natural enhancements of the primordial LP cutting plane algorithm (Algorithm 1) based on a semi-infinite LP formulation of the SDP.

The five algorithms are summarized in the table below:

Table I. Cutting plane algorithms for semidefinite programming

Name	Name	Model	Subproblem
Algorithm 1	LP cutting plane	Polyhedral	LP
Algorithm 2	Polyhedral bundle	Polyhedral	QP
Algorithm 3	SDP cutting plane	Non-polyhedral	SDP
Algorithm 4	Spectral bundle	Non-polyhedral	Quadratic SDP
Algorithm 5	Primal active set method for SDP	Non-polyhedral	SDP

We conclude with the following observations:

- Theorem 5 implies that solving an SDP is equivalent to solving an LP, provided we know the null space of optimal dual slack matrix S. This is a difficult requirement, and explains the poor convergence of Algorithm 1. Solving (LDR) and (LPR) approximately with an interior point method in Step 2 of Algorithm 1 does improve the performance of the algorithm somewhat. On the other hand, the non-polyhedral cutting plane algorithms, especially Algorithms 4 and 5, work around this requirement by identifying an appropriate subspace over which the dual slack matrix S is required to be psd. This is more practical than the stringent requirements of Theorem 5.

- The spectral bundle method (Algorithm 4) appears to the most efficient of all the algorithms in practice. As an aside we must mention that the computational aspects of Algorithm 5 have not been completely tested, and we hope that it will emerge as a serious competitor to the spectral bundle method in practice.

- On the other hand Algorithms 1 and 3 have proven polynomial time complexities, which compare well with the worst case complexity of interior point methods for SDP.

- It would be nice to have a cutting plane algorithm with proven worst case polynomial complexity that is also very efficient in practice.

References

F. Alizadeh, J.P.A. Haeberly and M.L. Overton, Complementarity and Nondegeneracy in Semidefinite Programming, *Mathematical Programming*, 77, 111-128, 1997.

E.W. Cheney and A.A. Goldstein, Newton's method for convex programming, *Numerische Mathematik*, 1, 253-268, 1959.

J.-L. Goffin and J.-P. Vial, Convex nondifferentiable optimization: a survey focussed on the analytic center cutting plane method, *Optimization Methods and Software*, 17, 805-867, 2002.

M. Grötschel, L. Lovasz and A. Schrijver, *Geometric Algorithms and Combinatorial Optimization*, Springer Verlag, Berlin, 1993.

C. Helmberg, http://www-user.tu-chemnitz.de/~helmberg/semidef.html *Semidefinite Programming Homepage*, 1996.

C. Helmberg, *Semidefinite Programming for Combinatorial Optimization*, Habilitationsschrift, ZIB-Report ZR-00-34, Konrad-Zuse-Zentrum Berlin, October 2000.

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

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

C. Helmberg and F. Rendl, A Spectral Bundle Method for Semidefinite Programming, *SIAM Journal on Optimization*, 10, 673–696, 2000.

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

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

R.A. Horn and C.R. Johnson, *Matrix Analysis*, Cambridge University Press, New York, 1990.

J.E. Kelley, The cutting plane method for solving convex programs, *Journal of the SIAM*, 8, 703-712, 1960.

K.C. Kiwiel, *Methods of Descent for Nondifferentiable Optimization*, Lecture Notes in Mathematics, 1133, Springer Verlag Berlin, 1985.

K. Krishnan, *Linear Programming Approaches for Semidefinite Programming Problems*, Ph.D. thesis, Dept. of Mathematical Sciences, Rensselaer Polytechnic Institute, July 2002.

K. Krishnan and J.E. Mitchell, A Linear Programming Approach to Semidefinite Programming Problems, Technical Report, Dept. of Mathematical Sciences, Rensselaer Polytechnic Institute, May 2001.

K. Krishnan and J.E. Mitchell, Primal active set approaches for semidefinite programming problems, under preparation.

C. Lemarechal, Nondifferentiable Optimization, in *Optimization*, (G.L. Nemhauser, A.H.G. Rinnooy Kan, and M.J. Todd, eds.), North-Holland, Amsterdam-New York-Oxford-Tokyo, 529–572, 1989.

M.M. Makela and P. Neittaanmaki, *Nonsmooth Optimization: Analysis and Algorithms with Applications to Optimal Control*, World Scientific Publishing Co., 1992.

J.E. Mitchell, Computational Experience with an Interior Point Cutting Plane Algorithm, *SIAM Journal on Optimization*, 10(4), 1212–1227, 2000.

J.E. Mitchell, Polynomial interior point cutting plane methods, Technical Report, Dept. of Mathematical Sciences, Rensselaer Polytechnic Institute, July 2001.

J.E. Mitchell and S. Ramaswamy, A Long-Step, Cutting Plane Algorithm for Linear and Convex Programming, *Annals of OR*, 99, 95–122, 2000.

M. Oskoorouchi, *The Analytic Center Cutting Plane Method with Semidefinite Cuts*, Ph.D. thesis, Faculty of Management, McGill University, Montreal, Canada, May 2002.

M. Oskoorouchi and J. L. Goffin, The Analytic Center Cutting Plane Method with Semidefinite Cuts, GERAD Technical Report G–2000–32, February 2002.

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

M.L. Overton, Large-Scale Optimization of Eigenvalues, *SIAM Journal on Optimization*, 2(1), 88–120, 1992.

G. Pataki, Cone-LP's and Semidefinite Programs: Geometry and a Simplex-type Method, *Proceedings of the Fifth IPCO Conference*, (W. H. Cunningham, S. T. McCormick, and M. Queyranne, eds.), Springer Verlag 1996.

G. Pataki, On the Rank of Extreme Matrices in Semidefinite Programs and the Multiplicity of Optimal Eigenvalues, *Mathematics of Operations Research*, 23(2), 339–358, 1998.

M.J. Todd, Semidefinite Optimization, *Acta Numerica* 10, 515-560, 2001.

L. Vandenberghe and S. Boyd, Semidefinite Programming, *SIAM Review*, 38, 49-95, 1996.

H. Wolkowicz, R. Saigal and L. Vandenberghe, *Handbook on Semidefinite Programming*, Kluwer Academic Publishers, 2000.