Optimization Methods and Software
Vol. 21, No. 1, February 2006, 57–74

Taylor & Francis
Taylor & Francis Group

A unifying framework for several cutting plane methods for semidefinite programming

KARTIK KRISHNAN*† and JOHN E. MITCHELL‡

†Department of Computing & Software, McMaster University, Hamilton, Ontario L8S 4K1, Canada
‡Mathematical Sciences, Rensselaer Polytechnic Institute, Troy, NY 12180, USA

(Received 12 December 2002; in final form 4 September 2004)

Cutting plane methods provide the means to solve large scale semidefinite programs (SDP) cheaply and quickly. They can also conceivably be employed for the purposes of re-optimization after branching or the addition of cutting planes. We give a survey of various cutting plane approaches for SDP in this paper. These cutting plane approaches arise from various perspectives, and include techniques based on interior point cutting plane approaches, non-differentiable optimization, and finally an approach which mimics the simplex method for linear programming (LP).

We present an accessible introduction to various cutting plane approaches that have appeared in the literature. We place these methods in a unifying framework which illustrates 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; Non-differentiable optimization; Interior point cutting plane methods; Active set approaches

1. Introduction

Semidefinite programming (SDP) has been one of the most exciting and active areas in optimization recently. Some excellent references for SDP include the survey papers by Todd [1] and Vandenberghe and Boyd [2], the SDP handbook edited by Wolkowicz *et al.* [3], and the web site maintained by Helmberg [4]. The tremendous activity in SDP was spurred by the discovery of efficient interior point algorithms for solving (SDP), and its important applications in control, in developing approximation algorithms for combinatorial optimization problems, finance, and statistics. However, these applications require effective techniques for solving large SDPs quickly. Although interior point algorithms are a great theoretical tool, they are fairly limited in the size of problems they can handle. Another drawback of interior point methods is that no good warm start techniques are available for re-optimization, after branching, or the addition of cutting planes. We discuss cutting plane approaches for SDP in this paper, which address these shortcomings. Our aim is to provide a unifying framework for the cutting plane approaches for SDP which have appeared in the literature.

*Corresponding author. Email: kartik@optlab.mcmaster.ca

Optimization Methods and Software
ISSN 1055-6788 print/ISSN 1029-4937 online © 2006 Taylor & Francis
http://www.tandf.co.uk/journals
DOI: 10.1080/10556780500065283

Consider the SDP problem

$$\begin{array}{ll} \min & C \cdot X \\ \text{s.t.} & \mathcal{A}(X) = b \\ & X \succeq 0, \end{array} \qquad \text{(SDP)}$$

with dual

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

where $X, S \in \mathcal{S}^n$ the space of real symmetric $n \times n$ matrices, $b \in \mathbb{R}^m$. Here m denotes the number equality constraints in the primal problem (SDP). Also,

$$C \cdot X = \sum_{i,\, j=1}^{n} C_{ij} X_{ij}$$

is the Frobenius inner product of matrices in $\mathcal{S}^n$. The linear operator

$$\mathcal{A}\colon \mathcal{S}^n \to \mathbb{R}^m,$$

and its adjoint

$$\mathcal{A}^{\mathrm{T}}\colon \mathbb{R}^m \to \mathcal{S}^n$$

are of the form:

$$\mathcal{A}(X) = \begin{bmatrix} A_1 \cdot X \\ \vdots \\ A_m \cdot X \end{bmatrix} \quad \text{and} \quad \mathcal{A}^{\mathrm{T}} y = \sum_{i=1}^{m} y_i A_i,$$

where the matrices $A_i, C \in \mathcal{S}^n$ are the given problem parameters. Here m denotes the number of primal constraints. 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 $d^{\mathrm{T}} X d \geq 0$, $\forall d \in \mathbb{R}^n$. On the other hand, $X \succ 0$ denotes a positive definite (pd) matrix, i.e., $d^{\mathrm{T}} X d > 0$, for all nonzero vectors $d \in \mathbb{R}^n$. Hereafter, we use $\mathcal{S}^n_+$ and $\mathcal{S}^n_{++}$ to denote the space of symmetric psd and pd, matrices, respectively. A good reference source is Horn and Johnson [5].

We will make the following assumptions.

ASSUMPTION 1 *The matrices A_i, $i = 1, \ldots, m$ are linearly independent in $\mathcal{S}^n$.*

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

Assumption 2 guarantees both (SDP) and (SDD) attain their optimal solutions X^* and (y^*, S^*), and their optimal values are equal, i.e., $C \cdot X^* = b^{\mathrm{T}} y^*$. Thus, the duality gap $X^* \cdot S^* = 0$ at optimality.

ASSUMPTION 3 *The equality constraints $\mathcal{A}(X) = b$ in (SDP) imply* $\operatorname{trace}(X) = a$, *for some constant $a \geq 0$.*

Assumption 3 enables one to rewrite (SDD) as an eigenvalue optimization problem and also ensures the existence of the following $\hat{y}$.

PROPOSITION 1 *Under Assumption* 3, *there exists a unique* $\hat{y} \in \mathbb{R}^m$ *satisfying*

$$\mathcal{A}^{\mathrm{T}}\hat{y} = I.$$

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

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 $A_i \cdot X = b_i$, $i = 1, \ldots, m$. Letting the components of $\hat{y}$ to be the coefficients in this linear combination we get the desired result. Uniqueness follows from Assumption 1. ■

We can write down the Lagrangian dual to (SDP) transferring all the equality constraints into the objective function via Lagrangian multipliers y_i, $i = 1, \ldots, m$, to give the following problem

$$\max_y b^{\mathrm{T}}y + \min_{X:\mathrm{trace}(X)=a,\ X\succeq 0} \left(C - \sum_{i=1}^m y_i A_i\right) \cdot X. \tag{1}$$

Assumption 2 ensures that this problem is equivalent to SDP. Using the variational characterization of the minimum eigenvalue function, the quantity in the inner minimization can be expressed as $a\lambda_{\min}(C - \mathcal{A}^{\mathrm{T}}y)$. We can then rewrite equation (1) as

$$\max_y b^{\mathrm{T}}y + a\lambda_{\min}(C - \mathcal{A}^{\mathrm{T}}y). \tag{2}$$

This is an eigenvalue optimization problem. We shall return to the formulation (2), 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. Consider the function

$$\begin{aligned} f(y) &= b^{\mathrm{T}}y + \lambda_{\min}(C - \mathcal{A}^{\mathrm{T}}y) \\ &= \lambda_{\min}\left(C - \sum_{i=1}^m y_i(A_i - b_i I)\right). \end{aligned}$$

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

$$\begin{aligned} \partial f(y) &= \mathrm{conv}\{b - \mathcal{A}(\,pp^{\mathrm{T}})\colon p = Px, x^{\mathrm{T}}x = 1\} \\ &= \{b - \mathcal{A}(PVP^{\mathrm{T}})\colon V \in \mathcal{S}^r,\ \mathrm{trace}(V) = 1,\ V \succeq 0\} \end{aligned} \tag{3}$$

where conv denotes the convex hull operation. The equivalence of the two expressions in (3) can be found in Overton [6]. Each member of $\partial f(y)$ is called a subgradient. The cutting plane approaches for SDP fall in the following categories:

- *Interior point cutting plane methods*: The SDP is a convex optimization problem with a weak polynomial time separation oracle [7], and hence can be solved within an interior

point cutting plane framework. Good surveys of such methods appear in refs. [8,9]. In particular, Algorithms 1 and 3 discussed in this survey fall within this framework.

- *Bundle methods for non-differentiable optimization*: An SDP with some additional restrictions (see Assumption 3) can be written as an eigenvalue optimization problem. These are convex but non-smooth optimization problems that can be handled by bundle methods for non-differentiable optimization. A survey on bundle methods appears in Lemarechal [25]. In particular, Algorithms 2 and 4 discussed in the survey fall in this class.
- *Active set approaches for SDP*: These approaches generate iterates which are on the boundary of the SDP feasible region. In particular, the simplex-like approach developed in Pataki [35] is a special case of such an approach. Algorithm 5 discussed in this survey belongs to this class.

We are interested in cutting plane methods which deal directly with the dual problem (SDD); in particular, the eigenvalue optimization problem (2). In this regard, at the time of writing, we are aware of at least four distinct cutting plane approaches namely: an linear programming (LP) cutting plane scheme for (SDD) due to Krishnan and Mitchell [10] (see, also, ref. [11]), a variant of analytic center cutting plane methods (ACCPM) incorporating semidefinite cuts due to Oskoorouchi and Goffin [12], the spectral bundle method due to Helmberg and Rendl [13], and a non-polyhedral primal active set approach due to Krishnan *et al.* [14]. These approaches are discussed in this survey.

We must also mention that there are two ACCPM schemes for SDP due to Toh *et al.* [15] and Sun *et al.* [16], but these deal with the SDP problem, with a large number of linear constraints, in the primal formulation. The methods use a cutting plane method to approximate the linear constraints, whereas the SDP constraint is always explicitly included in the relaxations. We will not discuss these approaches in this paper.

2. Cutting plane models for SDP

In this section, we shall discuss polyhedral and non-polyhedral cutting plane models for SDP. Consider the following *semi-infinite* formulation of (SDD).

$$\begin{aligned} \max \quad & b^{\mathrm{T}} y \\ \text{s.t.} \quad & dd^{\mathrm{T}} \cdot \mathcal{A}^{\mathrm{T}} y \leq dd^{\mathrm{T}} \cdot C \quad \forall ||d||_2 = 1. \end{aligned} \tag{LDM}$$

We consider (SDD) instead of (SDP) because this gives a problem with m variables. In contrast, a semi-infinite formulation of SDP would involve $\binom{n+1}{2} = n(n+1)/2 = O(n^2)$ variables.

Note that if $m = O(n^2)$ it may be advantageous to proceed in the following manner. Let $q = n(n+1)/2 - m$. It is possible to reformulate (SDP) as a semi-infinite programming problem in q variables. This is advantageous if q is smaller than m, in particular if $q = O(n)$. Let $\mathcal{B}$: $\mathcal{S}^n \to \mathbb{R}^q$ be the null space operator corresponding to $\mathcal{A}$, so the kernel of $\mathcal{B}^{\mathrm{T}}$ is exactly the range of $\mathcal{A}$. From Assumption 1, we can regard $\mathcal{B}$ as being composed of q linear functions, each represented by a matrix $B_i \in \mathcal{S}^n$, and these matrices are linearly independent in $\mathcal{S}^n$. Let X^0 be a feasible solution to $\mathcal{A}(X) = b$. The set of feasible solutions to $\mathcal{A}(X) = b$ is the set of all matrices of the form $X = X^0 - \mathcal{B}^{\mathrm{T}}(u)$ for some $u \in \mathbb{R}^q$. The problem (SDP) can then be written equivalently as

$$\begin{aligned} \min_{u,\,X} \quad & C \cdot X_0 - C \cdot \mathcal{B}^{\mathrm{T}}(u) \\ \text{s.t.} \quad & \mathcal{B}^{\mathrm{T}}(u) + X = X^0 \\ & X \succeq 0. \end{aligned}$$

This problem is exactly in the form of (SDD), so we can construct a linear programming relaxation (LPR) of it in the form (LDR) (see below) with q variables.

We consider a discretization of (LDM). Given a finite set of vectors $\{d_i, i = 1, \ldots, k\}$, we obtain the relaxation

$$\begin{aligned} \max \quad & b^{\mathrm{T}} y \\ \text{s.t.} \quad & d_i d_i^{\mathrm{T}} \cdot \mathcal{A}^{\mathrm{T}} y \leq d_i d_i^{\mathrm{T}} \cdot C \quad \text{for } i = 1, \ldots, k. \end{aligned} \tag{LDR}$$

The LP dual to (LDR) can be written

$$\begin{aligned} \min \quad & C \cdot \left(\sum_{i=1}^{k} x_i d_i d_i^{\mathrm{T}} \right) \\ \text{s.t.} \quad & \mathcal{A} \left(\sum_{i=1}^{k} x_i d_i d_i^{\mathrm{T}} \right) = b \\ & x \geq 0. \end{aligned} \tag{LPR}$$

The problem (LPR) is a constrained version of (SDP); this is reflected in the following proposition.

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

The optimality conditions for (SDP) can be summarized in the following theorem [17]:

THEOREM 1 *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 $Q^{\mathrm{T}} Q = I_r$, $R^{\mathrm{T}} R = I_{n-r}$, $Q^{\mathrm{T}} R = 0$, and Λ, Ω, diagonal matrices in $\mathcal{S}_+^r$, and $\mathcal{S}_+^{n-r}$, such that*

$$X = [Q \quad R] \begin{bmatrix} \Lambda & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} Q^{\mathrm{T}} \\ R^{\mathrm{T}} \end{bmatrix}, \text{ and} \tag{4}$$

$$S = [Q \quad R] \begin{bmatrix} 0 & 0 \\ 0 & \Omega \end{bmatrix} \begin{bmatrix} Q^{\mathrm{T}} \\ R^{\mathrm{T}} \end{bmatrix} \tag{5}$$

hold.

The diagonal matrices Λ, Ω contain the non-zero eigenvalues of X and S in the spectral decompositions (4) and (5), respectively. In addition, $P = [Q \quad R]$ is an orthogonal matrix that contains the common set of eigenvectors.

We get an upper bound on r, using Theorem 2, due to Pataki [18] (also see, ref. [17]), on the rank of extreme matrices X in SDP.

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

It follows that $\sqrt{2m}$ is an overestimate of the upper bound on the rank of at least one optimal solution. We now present the perfect constraints that are needed in the LP relaxations if the optimal solution to (SDP) is unique with distinct positive eigenvalues.

THEOREM 3 *Let $X^* = Q \Lambda Q^{\mathrm{T}}$ be an optimal solution to (SDP), and let q_i, $i = 1, \ldots, r$ be the columns of Q. If the constraints of (LDR) include the constraints $q_i q_i^{\mathrm{T}} \cdot \mathcal{A}^{\mathrm{T}} y \leq q_i q_i^{\mathrm{T}} \cdot C$*

for $i = 1, \ldots, r$ then any optimal solution x^* to (LDR) gives an optimal solution to (SDP) by taking $X = \sum_{i=1}^{k} x_i^* d_i d_i^{\mathrm{T}}$.

Proof Re-ordering the constraints if necessary, we can assume that $d_i = q_i$ for $i = 1, \ldots, r$. We have $X^* = Q \Lambda Q^{\mathrm{T}} = \sum_{i=1}^{r} \lambda_i q_i q_i^{\mathrm{T}}$, where $\lambda_i > 0$, $i = 1, \ldots, r$, and q_i, $i = 1, \ldots, r$ are the corresponding eigenvectors. This gives a feasible solution to (LPR) with $x_i = \lambda_i$ for $i = 1, \ldots, r$ and $x_i = 0$ otherwise. Further, this feasible solution is optimal, since (LPR) is a constrained version of (SDP). Thus the optimal values of (LPR) and (SDP) are identical, so any optimal solution to (LPR) gives an optimal solution to (SDP). ■

We note that (LPR) can be rewritten as

$$\begin{aligned} \min \quad & C \cdot (\mathrm{DMD}^{\mathrm{T}}) \\ \text{s.t.} \quad & A_j \cdot (\mathrm{DMD}^{\mathrm{T}}) = b_j \quad j = 1, \ldots, m \\ & M \succeq 0 \\ & M \text{ diagonal.} \end{aligned} \tag{6}$$

Here $M \in \mathcal{S}^m$ and $D \in \mathbb{R}^{n \times m}$ with jth column d_j. Theorem 3 suggests that if the columns of D contain eigenbases for all the strictly positive eigenvalues of X^*, then the solution to equation (6) is an exact solution to SDP. In other cases, a solution provides an upper bound on this objective value. This polyhedral cutting plane model is the basis for Algorithms 1 and 2 of section 4.

We can consider alternate non-linear models, which are simple extensions of this polyhedral cutting plane model in the following manner. Consider the following relaxation of equation (6) dropping the requirement that M be diagonal.

$$\begin{aligned} \min \quad & C \cdot (\mathrm{DMD}^{\mathrm{T}}) \\ \text{s.t.} \quad & A_j \cdot (\mathrm{DMD}^{\mathrm{T}}) = b_j \quad j = 1, \ldots, m \\ & M \succeq 0, \end{aligned} \tag{7}$$

with $M \in \mathcal{S}^m$ and $D \in \mathbb{R}^{n \times m}$. If $m = n$, then equation (7) is essentially SDP. In fact, if $\mathrm{Range}(D) \supset \mathrm{Range}(X^*)$, then a solution to equation (7) is an exact solution to SDP. This is a less stringent requirement than the polyhedral cutting plane model, where we require the exact eigenvectors of X^* and forms the basis of Algorithms 4 and 5 in section 5.

Finally, another relaxation of equation (6) which is more restrictive than equation (7) is the following.

$$\begin{aligned} \min \quad & C \cdot (\mathrm{DMD}^{\mathrm{T}}) \\ \text{s.t.} \quad & A_j \cdot (\mathrm{DMD}^{\mathrm{T}}) = b_j \quad j = 1, \ldots, m \\ & M \succeq 0 \\ & M \text{ block diagonal.} \end{aligned} \tag{8}$$

This is the cutting plane model employed in Algorithm 3 in section 5.

3. Generic cutting plane approach for SDP

Having considered the various cutting plane models, we now subsequently present the generic cutting plane approach. These approaches work directly with the eigenvalue optimization

model (2), i.e., we assume the redundant constraint $\text{trace}(X) = 1$ is added to the formulation (SDP). A generic cutting plane approach can be stated as follows:

1. Choose an initial point $\hat{y}$, an initial finite set $\mathcal{D} = \{D_i\}$, and a scalar $u \geq 0$.
2. Solve the following subproblem

$$\begin{aligned} \max \quad & \lambda + b^{\mathrm{T}} y - \frac{u}{2}||y - \hat{y}||^2 \\ \text{s.t.} \quad & D_i^{\mathrm{T}}(C - \mathcal{A}^{\mathrm{T}} y) D_i \succeq \lambda I, \quad i \in \mathcal{D} \end{aligned} \tag{9}$$

   to get (y^*, λ^*).
3. If $S^* = (C - \mathcal{A}^{\mathrm{T}} y^*) \succeq 0$, we are optimal; STOP. Else find $D \in \mathbb{R}^{n \times r}$, with $r \leq \sqrt{2m}$ such that $D^{\mathrm{T}}(C - \mathcal{A}^{\mathrm{T}} y^*) D \not\succeq 0$.
4. Either add D to $\mathcal{D}$ or aggregate D into $\mathcal{D}$.
5. Set $\hat{y} = y^*$, and return to step 2.

The five methods to be discussed in sections 4 and 5 can then be distinguished as in table 1.

Note that the variable λ in equation (9) corresponds to the redundant constraint $\text{trace}(X) = 1$.

Algorithms 1 and 2 are polyhedral because only $D \in \mathbb{R}^{n \times 1}$ appear in $\mathcal{D}$ so the constraints are actually linear inequalities of the form $D_i^{\mathrm{T}}(C - \mathcal{A}^{\mathrm{T}} y) D_i \geq 0, \forall i$. Likewise, Algorithm 3 is non-polyhedral, because not all the constraints $D_i^{\mathrm{T}}(C - \mathcal{A}^{\mathrm{T}} y) D_i \succeq 0$ can be reduced to a finite set of linear inequalities. Finally, Algorithms 4 and 5 have just one semidefinite block of constraints.

For Algorithms 1, 3 and 5, the weight $u = 0$. In these cases, the dual to the subproblem being solved in each iteration is

$$\begin{aligned} \min \quad & C \cdot \left(\sum_{i \in \mathcal{D}} D_i V_i D_i^{\mathrm{T}} \right) \\ \text{s.t.} \quad & \mathcal{A} \left(\sum_{i \in \mathcal{D}} D_i V_i D_i^{\mathrm{T}} \right) = b \\ & \sum_{i \in \mathcal{D}} \text{trace}(V_i) = 1 \\ & V_i \succeq 0, \quad i \in \mathcal{D}. \end{aligned}$$

This takes the forms (6), (7) and (8), which correspond to Algorithms 1, 3 and 5, respectively. Hence, these algorithms are solving a relaxation of (SDD) in every iteration.

Table 1. Cutting plane methods for SDP.

Algorithm	Model	u	Form of $\mathcal{D}$	Add or aggregate?
1	Polyhedral	Zero	$D \in \mathbb{R}^{n \times 1}$ only	Add, letting $\lvert\mathcal{D}\rvert$ grow
2	Polyhedral bundle	Positive	$D \in \mathbb{R}^{n \times 1}$ only	Add, letting $\lvert\mathcal{D}\rvert$ grow
3	Non-linear block-diag	Zero	$D \in \mathbb{R}^{n \times r}$ with $r \leq \sqrt{2m}$	Add, letting $\lvert\mathcal{D}\rvert$ grow
4	Spectral bundle	Positive	$D \in \mathbb{R}^{n \times r}$	Aggregate, keeping $\lvert\mathcal{D}\rvert = 1$
5	Primal active set	Zero	$D \in \mathbb{R}^{n \times r}$	Aggregate, keeping $\lvert\mathcal{D}\rvert = 1$

For Algorithms 2 and 4, the weight $u > 0$, since the proximal bundle idea is being used. In this case, the Lagrangian dual to the subproblem is a quadratic SDP and has the form

$$\begin{aligned} \min \quad & \frac{1}{2u}\left\| b - \mathcal{A}\left(\sum_{i\in\mathcal{D}} D_i V_i D_i^{\mathrm{T}}\right)\right\|^2 - b^{\mathrm{T}}\hat{y} \\ & -(C - \mathcal{A}^{\mathrm{T}}\hat{y}) \cdot \left(\sum_{i\in\mathcal{D}} D_i V_i D_i^{\mathrm{T}}\right) \\ \text{s.t.} \quad & \sum_{i\in\mathcal{D}} \operatorname{trace}(V_i) = 1 \\ & V_i \succeq 0, \quad i \in \mathcal{D} \end{aligned}$$

Owing to proximal bundle term in Algorithms 2 and 4, these algorithms may not always be solving a relaxation of (SDD).

4. Polyhedral cutting plane algorithms for SDP

We present two polyhedral cutting plane algorithms for SDP in this section. A discussion on Algorithm 1 appears in section 4.1, while Algorithm 2 appears in section 4.2. Both these algorithms are based on the polyhedral model (6).

4.1 *Algorithm 1: polyhedral cutting plane algorithm*

The method is originally due to Cheney and Goldstein [19] and Kelley [20]. It was introduced by Krishnan and Mitchell [10] and Goldfarb [11] in the context of SDP. The complete algorithm appears in figure 1.

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

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. Choose an appropriate approximate optimality criterion for the solution to (LPR) and (LDR).
2. In the kth iteration, obtain a solution y^k to the discretization (LDR) and its dual (LPR) satisfying the approximate optimality criterion. Update the upper bound: UB $= \min\{\text{UB}, b^{\mathrm{T}} y^k\}$.
3. Compute $\lambda = |\lambda_{\min}(C - \mathcal{A}^{\mathrm{T}} y^k)|$, and a corresponding eigenvector d. Update the lower bound: $LB = \max\{\text{LB}, b^{\mathrm{T}} y^k + \lambda a\}$, where a is as in Assumption 3. If $|\text{LB} - \text{UB}| \leq \epsilon_1$ or $\lambda \leq \epsilon_2$, go to step 5.
4. Add the constraint
$$dd^{\mathrm{T}} \bullet \mathcal{A}^{\mathrm{T}} y \leq dd^{\mathrm{T}} \bullet C$$
to (LDR). Set $k = k + 1$, update the approximate optimality criterion for (LDR) and (LPR) if desired and return to step 2.
5. The current solution (x^k, y^k) for (LDR) and (LPR) give an optimal solution (X, y) for (SDP), and (SDD) respectively.

Figure 1. Algorithm 1: polyhedral cutting plane algorithm.

There are several choices for an appropriate approximate optimality criterion for the solution of (LDR) and (LPR) in step 2 of Algorithm 1. The simplest is to require that the linear programs are solved to optimality, i.e., to within a relative duality gap of 10^{-8}, at each stage; in this case, the query point y^k for the oracle is a solution to (LDR). Alternatively, a dynamically modified tolerance on the duality gap could be used, with the tolerance being tightened as k increases, so that an optimal solution to SDP is eventually obtained. There are two noteworthy advantages to solving the relaxations approximately using an interior point cutting plane method: there is less oscillation in the sequence of iterates, and the cutting planes are generated at points that are more central so they tend to be deeper.

The algorithm employs the weak polynomial separation oracle for SDP in step 3. Typically, the most negative eigenvalue $\lambda_{\min}(S)$, and its associated eigenvector are estimated by an iterative method like the Lanczos scheme. This oracle can be implemented in $O(n^3 + mn^2)$ arithmetic operations [21,22]. When the algorithm converges, both optimality criteria in step 3 will eventually be satisfied; one may be preferred to the other in certain situations.

There is another way to motivate this cutting plane approach, which is based on the eigenvalue optimization model (2). Assume that we have a set of points $y = y^1, \ldots, y^k$, and we know the function values $f(y^i)$, $i = 1, \ldots, k$, and subgradients $(b - \mathcal{A}(d_i d_i^{\mathrm{T}}))$, $i = 1, \ldots, k$ (where d_i is a normalized eigenvector corresponding to $\lambda_{\min}(C - \mathcal{A}^{\mathrm{T}} y^i)$) at these points. We can construct the following overestimate $\hat{f}_m(y)$ for $f(y)$.

$$\hat{f}_m(y) = \min_{i=1,\ldots,k} d_i d_i^{\mathrm{T}} \cdot (C - \mathcal{A}^{\mathrm{T}} y) + b^{\mathrm{T}} y \geq f(y).$$

To see that $\hat{f}_m(y)$ is an overestimate note that since the d_i are normalized, we have

$$\begin{aligned} \lambda_{\min}(C - \mathcal{A}^{\mathrm{T}} y) &\leq d_i^{\mathrm{T}} (C - \mathcal{A}^{\mathrm{T}} y) d_i, \quad i = 1, \ldots, k \\ &= d_i d_i^{\mathrm{T}} \cdot (C - \mathcal{A}^{\mathrm{T}} y), \quad i = 1, \ldots, k \end{aligned}$$

We now maximize this overestimate instead, i.e.,

$$\max_y \hat{f}_k(y) = \max_y \{b^{\mathrm{T}} y + \min_{i=1,\ldots,k} \{d_i d_i^{\mathrm{T}} \cdot (C - \mathcal{A}^{\mathrm{T}} y)\}\},$$

which can be recast as the following linear program

$$\begin{aligned} \max \quad & b^{\mathrm{T}} y + v \\ \text{s.t.} \quad & d_i d_i^{\mathrm{T}} \cdot \mathcal{A}^{\mathrm{T}} y + v \leq d_i d_i^{\mathrm{T}} \cdot C, \quad i = 1, \ldots, k, \end{aligned} \tag{10}$$

with dual

$$\begin{aligned} \min \quad & C \cdot \left(\sum_{i=1}^{k} x_i d_i d_i^{\mathrm{T}} \right) \\ \text{s.t.} \quad & \mathcal{A} \left(\sum_{i=1}^{k} x_i d_i d_i^{\mathrm{T}} \right) = b_j, \quad j = 1, \ldots, m \\ & \sum_{i=1}^{k} x_i = 1 \\ & x \geq 0. \end{aligned} \tag{11}$$

This is exactly the problem obtained by considering a discretization of (SDD). Here, v is the dual variable corresponding to the redundant constraint $\sum_{i=1}^{k} x_i = 1$, which is implicitly satisfied by any solution x to (LPR). Thus, we can set $v = 0$ without any loss of generality. The solution (v, y) with $v = 0$ is the one corresponding to (LDR).

Unfortunately, Algorithm 1 has a very poor rate of convergence in practice. For instance, we observed that a simplex implementation that solved each linear program to optimality performed very badly [10,21]. Here, the query points y^k are extreme points of the constraint set for (LDR). Primarily, minimizing $\hat{f}_k$ to find y^{k+1} makes sense only if $\hat{f}_k \approx f$, near y^k, this is one of the reasons for the slow convergence for the cutting plane scheme. Lemarechal [25] discusses some convergence estimates for such an algorithm.

An alternative is to solve a feasibility problem in lieu of (LDR) in step 2 of Algorithm 1; here y^k is the approximate analytic or volumetric center for the polyhedron that includes the constraints and multiple copies of a constraint corresponding to the objective function in (LDR). These centers can be found using interior point algorithms, and the approaches are analogues to the ellipsoid algorithm [7], where the new iterate y^k is the updated center of the new ellipsoid. A formal discussion of such approaches including termination criteria and the definitions of analytic and volumetric centers can be found in the surveys by Krishnan and Terlaky [23] and Mitchell [9]. In this case, the number of copies of the objective function constraint and the depth of this constraint are modified in order to push the solution to optimality as k is increased. The volumetric center algorithm for the convex feasibility problem [24] is such an approach; using this framework for the cutting plane approach of Algorithm 1 gives an algorithm that requires $O(m \log(1/\epsilon))$ calls to the oracle and $O(m^4 \log(1/\epsilon))$ other arithmetic operations. The overall complexity is better than employing the ellipsoid method for SDP and also compares favourably with interior point methods for SDP [21,22].

4.2 *Algorithm 2: polyhedral bundle scheme*

One way to improve the convergence of the algorithm is to utilize the proximal bundle idea discussed in Lemarechal [25], Kiwiel [26], Hiriart-Urruty and Lemarechal [27], and Makela and Neittaanmaki [28]. This leads naturally to Algorithm 2.

Before discussing the actual algorithm, we present a short discussion on the proximal bundle scheme. The rough idea here is to maximize $\hat{f}_k(y) - u/2||y - y^k||^2$ (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^k. 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 [25] remarks, choosing this parameter u is an art in itself. The regularization penalty term $\frac{u}{2}||y - y^k||^2$ acts as a trust region constraint $||y - y^k||^2 \leq \sigma_k$ 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.

Consider adding this quadratic term in the objective function of equation (10) giving

$$\begin{aligned} \max \quad & b^{\mathrm{T}} y + v - \frac{u}{2}||y - \hat{y}||^2 \\ \text{s.t.} \quad & v \leq d_i d_i^{\mathrm{T}} \cdot (C - \mathcal{A}^{\mathrm{T}} y), \quad i = 1, \ldots, k, \end{aligned} \tag{12}$$

with Lagrangian dual

$$\begin{aligned} \min \quad & \frac{1}{2u}||b - \mathcal{A}(X))||^2 - b^{\mathrm{T}}\hat{y} \\ & -(C - \mathcal{A}^{\mathrm{T}}\hat{y}) \cdot X \\ \text{s.t.} \quad & X = \sum_{i=1}^{k} x_i d_i d_i^{\mathrm{T}} \\ & \sum_{i=1}^{k} x_i = 1 \\ & x_i \geq 0, \quad i = 1, \ldots, k. \end{aligned} \tag{13}$$

Setting $u = 0$ in equation (12) gives equation (10). Owing to strong duality, equations (12) and (13) have the same objective value. Their solutions y and X satisfy

$$y = \hat{y} + \frac{1}{u}(b - \mathcal{A}(X)). \tag{14}$$

The complete algorithm appears in figure 2. The formal proof of convergence of the algorithm can be found in Kiwiel [29].

Let

$$\hat{W} = \left\{ \sum_{i=1}^{k} x_i d_i d_i^{\mathrm{T}} \,|\, \sum_{i=1}^{k} x_i = 1, x_i \geq 0, i = 1, \ldots, k \right\}, \tag{15}$$

which is the feasible region of equation (13). It appears that the size of $\hat{W}$ grows indefinitely with iteration count in the earlier algorithm. We can, however, choose to keep the number of subgradients no larger than an a priori bound l. We retain the earlier $l - 2$ subgradients $d_i d_i^{\mathrm{T}}$ corresponding to the largest values of x, along with the new subgradient $p^{k+1} p^{(k+1)^{\mathrm{T}}}$ in the

1. Let $y^1 \in \mathbb{R}^m$, let $p^1 \in \mathbb{R}^n$ be a normalized eigenvector corresponding to $\lambda_{\min}(C - \boldsymbol{\mathcal{A}}^{\mathrm{T}} y^1)$. Also choose the weight $u > 0$, an improvement parameter $\nu \in (0, 1)$ and finally a termination parameter $\epsilon > 0$.
2. At iteration k, compute X^{k+1} from equation (13) and y^{k+1} from equation (14), where $\hat{y} = y^k$. Also, let $f_{X^{k+1}}(y^{k+1}) = b^{\mathrm{T}} y^{k+1} + (C - \boldsymbol{\mathcal{A}}^{\mathrm{T}} y^k) \bullet X^{k+1}$.
3. If
$$f_{X^{k+1}}(y^{k+1}) - f(\hat{y}^k) \leq \epsilon$$
stop.
4. Compute $f(y^{k+1})$, and the eigenvector p^{k+1} corresponding to $\lambda_{\min}(C - \boldsymbol{\mathcal{A}}^{\mathrm{T}} y^{k+1})$.
5. If the actual increase is not much smaller than the increase predicted by the model (sufficient increase), i.e.,
$$f(y^{k+1}) - f(\hat{y}^k) \geq \nu(f_{X^{k+1}}(y^{k+1}) - f(\hat{y}^k))$$
then perform a serious step, i.e., $\hat{y}^{k+1} = y^{k+1}$. Else, perform a null step, i.e., $y^{k+1} = y^k$.
6. Return to step (2).

Figure 2. Algorithm 2: polyhedral bundle method for (SDP).

bundle and aggregate the rest of the subgradients in a subgradient matrix $\bar{W}^l$. The set $\hat{W}^{k+1}$ is then the convex hull of $\bar{W}^l$ and the $l-1$ subgradients in the bundle, i.e.,

$$\hat{W}^{k+1} = \left\{ \alpha \bar{W}^l + \sum_{i=1}^{l-1} x_i d_i d_i^{\mathrm{T}} | \alpha + \sum_{i=1}^{l-1} x_i = 1, \alpha \geq 0, x_i \geq 0, i = 1, \ldots, r \right\}, \tag{16}$$

and so the feasible region of equation (13) is modified to require that X be in this set $\hat{W}^{k+1}$. Alternatively, in the kth iteration, we can choose $\hat{W}^{k+1}$ to be the convex hull of X^{k+1} and $p^{k+1} p^{(k+1)^{\mathrm{T}}}$, in which case $l = 2$. It can be shown that the algorithm converges in these situations too.

5. Non-polyhedral cutting plane algorithms for SDP

In this section, we discuss three non-polyhedral cutting plane algorithms for SDP. The three algorithms appear in sections 5.1, 5.2, and 5.3, respectively. The first algorithm is based on the non-polyhedral block diagonal model (8), whereas the latter two schemes work with the model (7).

5.1 *Algorithm 3: non-polyhedral block diagonal cutting plane scheme*

The next algorithm we consider is based on the non-polyhedral, block-diagonal SDP model. This is employed in Oskoorouchi and Goffin [12]; see also Oskoorouchi [30]. This algorithm is identical to Algorithm 1 except when the multiplicity of $\lambda_{\min}(C - \mathcal{A}^{\mathrm{T}} y)$ is greater than one. Since we are essentially minimizing this quantity in (SDD), during the course of Algorithm 1, the smaller eigenvalues generally tend to coalesce together thereby increasing the multiplicity of this eigenvalue. In fact at optimality, this multiplicity is bounded by $\sqrt{2m}$ (from Theorems 1 and 2).

When this number is r (say), we could instead add the following semidefinite constraint

$$\sum_{i=1}^{m} y_i (D^{\mathrm{T}} A_i D) \preceq (D^{\mathrm{T}} C D),$$

where $D \in \mathbb{R}^{n \times r}$, with $D^{\mathrm{T}} D = I_r$, whose columns form an eigenbasis for the eigenspace of $C - \mathcal{A}^{\mathrm{T}} y$ with eigenvalue $\lambda_{\min}(C - \mathcal{A}^{\mathrm{T}} y)$. We note that this is much stronger than

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

This leads to the following subproblem to be solved in every iteration.

$$\begin{aligned} \max \quad & b^{\mathrm{T}} y \\ \text{s.t.} \quad & \sum_{i=1}^{m} y_i (D_j^{\mathrm{T}} A_i D_j) \preceq (D_j^{\mathrm{T}} C D_j), \quad j = 1, \ldots, k, \end{aligned} \tag{17}$$

whose dual is

$$\begin{aligned} \min \quad & C \cdot \left(\sum_{i=1}^{k} D_i V_i D_i^{\mathrm{T}} \right) \\ \text{s.t.} \quad & \mathcal{A} \left(\sum_{i=1}^{k} D_i V_i D_i^{\mathrm{T}} \right) = b \\ & V_i \succeq 0, \quad i = 1, \ldots, k, \end{aligned} \tag{18}$$

where the number of columns in D_j, $j = 1, \ldots, k$ reflect the multiplicities of $\lambda_{\min}(C - \mathcal{A}^{\mathrm{T}} y^i)$, $i = 1, \ldots, k$ and some of these could conceivably be 1.

The entire algorithm is detailed in figure 3.

Instead of solving equation (17) in step 2 of Algorithm 3, one can choose the query point y^k to be the approximate analytic center of the convex set that includes the feasible region of equation (17) and multiple copies of the objective function for this problem as inequality constraints. The ACCPM of Oskoorouchi and Goffin [12] is such an approach, which solves SDP in fully polynomial time; in particular, this can be done in $O(r^2 m^3/\mu^2 \epsilon^2)$ calls to the oracle, where r is an upper bound on the number of columns in D_i, $\mu > 0$ is a condition number on the cuts, and $\epsilon > 0$ is the tolerance to which one wants to solve the SDP.

5.2 *Algorithm 4: spectral bundle scheme*

We now discuss the spectral bundle method for SDP due to Helmberg and Rendl [13]. Other references include Helmberg [31], Helmberg and Kiwiel [32], Helmberg and Oustry [33], and Oustry [34].

We will motivate the spectral bundle scheme in this section, by considering the second aggregation scheme employed by the polyhedral bundle method of section 4.2, as in equation (16). The spectral bundle instead chooses the following expression for $\hat{W}$:

$$\hat{W} = \{\alpha \bar{W} + P V P^{\mathrm{T}} | \alpha + \mathrm{trace}(V) = 1, \alpha \geq 0, V \succeq 0\},$$

where $P \in \mathbb{R}^{n \times r}$, with $P^{\mathrm{T}} P = I_r$, whose columns are the p_i, $i = 1, \ldots, r$. The columns of the matrix P constitute the bundle. Here r refers to the size of the bundle, and inspired by

1. Choose an initial set of constraints for equation (17); this can be done in the same way as Algorithm 1. Choose an appropriate approximate optimality criterion for the solution to equations (17) and (18). Set $k = 1$.
2. Find an approximate solution (X^k, y^k) to equations (17) and (18).
3. Compute $\lambda_{\min}(C - \mathcal{A}^{\mathrm{T}} y^k)$ and an orthonormal matrix $D^k \in \mathbb{R}^{n \times r^k}$, where r^k is the multiplicity of this eigenvalue. Update the lower and upper bounds as in Algorithm 1. If $\lambda_{\min}(C - \mathcal{A}^{\mathrm{T}} y^k)$ is small, or the difference in bounds is small, go to step 4. Else, add the following constraint to equation (17)

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

   Set $k = k + 1$, update the approximate optimality criterion if desired and return to step 2.
4. The current solution (X^k, y^k) is optimal for (SDP), and (SDD) respectively.

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

Theorem 2, this number is chosen to be no greater than $\sqrt{2m}$. It is clear from these two expressions for $\hat{W}$ that the spectral bundle method does a better job in approximating the subdifferential of $\lambda_{\min}(S)$.

The spectral bundle method solves the following pair of subproblems in every iteration.

$$\begin{aligned} \max \quad & \lambda + b^{\mathrm{T}} y \\ & -\frac{u}{2}||y - \hat{y}||^2 \\ \text{s.t.} \quad & P^{\mathrm{T}}(C - \mathcal{A}^{\mathrm{T}} y) P \geq \lambda I \\ & (C - \mathcal{A}^{\mathrm{T}} y) \cdot \bar{W} \geq \lambda, \end{aligned} \tag{19}$$

with dual

$$\begin{aligned} \min \quad & \frac{1}{2u}||b - \mathcal{A}(X)||^2 - (C - \mathcal{A}^{\mathrm{T}} y^k) \cdot X - b^{\mathrm{T}} \hat{y} \\ \text{s.t.} \quad & X = \alpha \bar{W} + P V P^{\mathrm{T}} \\ & \alpha + \mathrm{trace}(V) = 1 \\ & \alpha \geq 0 \\ & V \succeq 0. \end{aligned} \tag{20}$$

The problem (20) is a quadratic (SDP) and can be solved efficiently for X using interior point methods, if r is small. More details can be found in Helmberg and Rendl [13] and Helmberg [31]. The solution y to equation (19) is then given by

$$y = \hat{y} + \frac{1}{u}(b - \mathcal{A}(X)). \tag{21}$$

The algorithm can be found in figure 4.

We present a short discussion on update rules for $\bar{W}$ and P. Other updates are possible, and these are discussed in length in Helmberg [31]. In the kth iteration, one solves (20) for V^k and computes its spectral decomposition $V^k = Q \Lambda Q^{\mathrm{T}}$. We then 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

1. Start with $y^1 \in \mathbb{R}^m$, let $p^1 \in \mathbb{R}^n$ a normalized eigenvector corresponding to $\lambda_{\min}(C - \mathcal{A}^{\mathrm{T}} y^1)$. Choose parameters $u > 0$, an improvement parameter $\nu_1 \in (0, 1)$, and a termination parameter $\epsilon > 0$. Let $P^1 = p^1$, and $\bar{W}^1 = p^1 p^{1\mathrm{T}}$.
2. In the kth iteration, solve equations (19) and (20) for (α^{k+1}, V^{k+1}), with $X^{k+1} = \alpha^{k+1} \bar{W}^k + P^k V^{k+1} P^{k\mathrm{T}}$. Compute y^{k+1} from equation (21). Let $f_{X^{k+1}}(y^{k+1}) = b^{\mathrm{T}} y^{k+1} + (C - \mathcal{A}^{\mathrm{T}} y^{k+1}) \bullet X^{k+1}$.
3. If $f(y^{k+1}) - f(\hat{y}^k) \leq \epsilon$, then stop with optimality.
4. Compute p^{k+1} a normalized eigenvector corresponding to $\lambda_{\min}(C - \mathcal{A}^{\mathrm{T}} y^{k+1})$.
5. If
$$f(y^{k+1}) - f(\hat{y}^k) \geq \nu(f_{X^{k+1}}(y^{k+1}) - f(\hat{y}^k))$$
then set $\hat{y}^{k+1} = y^{k+1}$. Else $\hat{y}^{k+1} = \hat{y}^k$.
6. Update the parameters of set $\hat{W}^{k+1}$ using equation (22). Set $k = k + 1$, and return to step 2.

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

of V^k, respectively. This distinction is based on keeping the size of the bundle P below the earlier mentioned bound of $\sqrt{2m}$. Finally, P and $\bar{W}$ are updated as follows:

$$\begin{aligned} P^{k+1} &= \text{orth}([P^k Q_1, v^{k+1}]), \\ \bar{W}^{k+1} &= \frac{1}{\alpha^k + \text{trace}(\Lambda_2)}(\alpha^k \bar{W}^k + P^k Q_2 \Lambda_2 (P^k Q_2)^{\mathrm{T}}). \end{aligned} \tag{22}$$

The proof of convergence of the spectral bundle scheme is along the same lines as the polyhedral bundle method and can be found in Helmberg and Rendl [13], and Helmberg [31]. In the extreme case of aggregation, i.e., when there are two subgradients in the bundle, as considered in section 4.2, the two methods are exactly the same. The spectral bundle method is only a first order method, but variants of the proximal bundle scheme which converge, and enjoy asymptotically a quadratic rate of convergence, were recently developed by Oustry [34].

5.3 *Algorithm 5: primal active set approach*

The final algorithm is a primal active set approach due to Krishnan *et al.* [14], which solves an SDP as a sequence of smaller SDPs in an active set framework. The method relies on the notions of extreme point solutions and non-degeneracy in SDP. The notion of extreme point solutions in SDP, and a crossover algorithm to generate them can be found in Pataki [35]. A good overview of non-degeneracy in the context of SDP can be found in Alizadeh *et al.* [17], whereas an alternative characterization appears in Pataki [36].

The method generates extreme point solutions X. The ranks r of these extreme point solutions satisfy the inequality $r(r+1)/2 \leq m$ (Theorem 2). There are two steps performed in each iteration, analogous to the primal simplex method for LP.

1. Construct a complementary dual solution (y, S). If $S \succeq 0$, then we are optimal.
2. Else if the previous extreme point solution was non-degenerate, we construct another extreme point solution, whose objective value is strictly lower than the previous one.

Given an extreme point iterate $X = [P_1 \quad P_2] \begin{bmatrix} \Lambda & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} P_1^{\mathrm{T}} \\ P_2^{\mathrm{T}} \end{bmatrix}$ of rank r, a complementary dual solution (y, S) satisfies $XS = 0$. This requires

$$\begin{aligned} \bar{S}_{11} &:= P_1^{\mathrm{T}}(C - \mathcal{A}^{\mathrm{T}} y) P_1 = 0, \\ \bar{S}_{12} &:= P_1^{\mathrm{T}}(C - \mathcal{A}^{\mathrm{T}} y) P_2 = 0. \end{aligned} \tag{23}$$

Setting the symmetric matrix $\bar{S}_{11}$ to zero gives $r(r+1)/2$ equations in y and requiring $\bar{S}_{12} = 0$ gives another $r(n-r)$ equations. If X is non-degenerate, then the coefficient matrix in equation (23) has full column rank, so it is typically overdetermined. Since X is also an extreme point, the first $r(r+1)/2$ equations in equation (23) are linearly independent. Since $m \geq r(r+1)/2$, one can choose m linearly independent equations from equation (23), including the first $r(r+1)/2$. The resulting system is hereafter denoted as $\bar{S}_B = 0$, whose unique solution is y^*. We update P to include all the columns p_j present in $\bar{S}_B = 0$; in particular, this contains all the columns in P_1. If $S^* = (C - \mathcal{A}^{\mathrm{T}} y^*)$ is not yet psd, we also add the eigenvector corresponding to $\lambda_{\min}(S^*)$, suitably orthonormalized, to P.

1. Consider an extreme point solution $X^1 = P^1 V^1 P^{1\mathrm{T}}$ with $V^1 \succ 0$.
2. In the kth iteration, choose a subset of m linearly independent equations from equation (23). Solve the resulting system $\bar{S}_B = 0$ for a unique y^{k+1}.
3. If $S^{k+1} = (C - \mathcal{A}^{\mathrm{T}} y^{k+1}) \succeq 0$, stop; else update $\bar{P}^{k+1}$ to be all the columns in P_1^k, and those in P_2^k included in $\bar{S}_B = 0$. Compute the normalized eigenvector p^{k+1} corresponding to $\lambda_{\min}(S^{k+1})$, and set $\bar{P}^{k+1} = \mathrm{orth}\,|\bar{P}^{k+1}, p^{k+1}|$. Update the lower and upper bounds as discussed in Algorithm 1. If the difference between these bounds is small, stop.
4. Solve equation (24), with $P = \bar{P}^{k+1}$. Let $V^{k+1} = {R_1}^{k+1} M^{k+1} {R_1^{k+1}}^{\mathrm{T}}$ with $M^{k+1} \succ 0$. Set ${P_1}^{k+1} = \bar{P}^{k+1} {R_1}^{k+1}$ and $X^{k+1} = {P_1}^{k+1} M^{k+1} {P_1}^{k+1\mathrm{T}}$.
5. If X^{k+1} is not an extreme point, run the crossover algorithm on X^{k+1} and return to step 1.

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

The update of X is based on the non-polyhedral cutting plane model (7). The new $X = \mathrm{PVP}^{\mathrm{T}}$ is obtained by solving the following pair of subproblems.

$$\begin{aligned} \min \quad & (P^{\mathrm{T}} C P) \cdot V \\ \text{s.t.} \quad & (P^{\mathrm{T}} A_i P) \cdot V = b_i, \quad i = 1, \ldots, m \\ & V \succeq 0, \end{aligned} \tag{24}$$

with dual

$$\begin{aligned} \max \quad & b^{\mathrm{T}} y \\ \text{s.t.} \quad & P^{\mathrm{T}}(C - \mathcal{A}^{\mathrm{T}} y) P \succeq 0. \end{aligned} \tag{25}$$

If the resulting X is not an extreme point, one can use the crossover algorithm in ref. [35] (Algorithm 1 in section 4.1) to generate an extreme point iterate, whose objective value is no worse than X.

The proof for strict decrease under non-degeneracy conditions can be found in Krishnan *et al.* [14]. The columns of P contain bases for the positive eigenspaces of the extreme point iterates X. This constitutes the active set in the algorithm and is analogous to the basis matrix in the simplex method for LP. Once the active set contains the eigenspace of the optimal solution X^*, the algorithm terminates.

The complete algorithm appears in figure 5. The size of P grows in step 3 and diminishes in steps 4 and 5 of the algorithm. The latter two steps can be regarded as constituting the necessary aggregation in the algorithm. We must mention that the convergence and computational aspects of Algorithm 5 are currently under investigation.

6. Conclusions

We present an accessible and unified introduction to various cutting plane methods that have appeared in the literature. These five methods are all roughly solving relaxations of the dual semidefinite program (SDD). Each of the algorithms arise as natural enhancements of the primordial LP cutting plane algorithm (Algorithm 1).

Algorithm 1 can be implemented in polynomial time, if one employs the volumetric center method to approximately solve the LP relaxations. In fact, this complexity compares favourably with interior point methods for SDP. Similarly, Algorithm 3 can be implemented in the ACCPM framework with SDP cuts in fully polynomial time. Some computational results

with Algorithm 1 can be found in Krishnan [21] and Krishnan and Mitchell [10]. We have also employed a variant of Algorithm 1 in a cut and price algorithm [37] designed to solving the maxcut problem to optimality.

On the other hand, the spectral bundle method (Algorithm 4) appears to be the most efficient of all the algorithms described in this survey. Excellent computational results have been obtained using the method [13,38] for problems that are inaccessible to IPMs due to the high demand for computer time and storage requirements.

The primal active set approach (Algorithm 5) mimics the primal simplex method for LP, and, like the dual simplex method for LP, dual variants of these approaches could be used for re-optimization after the addition of cutting planes. The computational performance of this algorithm is currently under investigation.

Highlighting these conclusions, one can say that there are variants of cutting plane methods with good polynomial complexity, warm start capabilities, and ones that are very efficient in practice especially on large SDPs.

Acknowledgements

The authors would like to thank the editor and three anonymous referees whose detailed comments greatly improved the presentation of the paper. The work was done as part of the first authors Ph.D. dissertation at RPI. The first author would also like to acknowledge financial support from Rice and McMaster Universities during his stints as a post-doctoral fellow there. Research supported in part by NSF grant numbers CCR–9901822 and DMS–0317323.

References

[1] Todd, M.J., 2001, Semidefinite optimization. *Acta Numerica*, **10**, 515–560.
[2] Vandenberghe, L. and Boyd, S., 1996, Semidefinite programming. *SIAM Review*, **38**, 49–95.
[3] Wolkowicz, H., Saigal, R. and Vandenberghe, L., 2000, *Handbook on Semidefinite Programming* (Kluwer Academic Publishers).
[4] Helmberg, C., 1996, Semidefinite Programming Homepage. Available online at: http://www-user.tu-chemnitz.de/~helmberg/semidef.html.
[5] Horn, R.A. and Johnson, C.R., 1990, *Matrix Analysis* (New York: Cambridge University Press).
[6] Overton, M.L., 1992, Large-scale optimization of eigenvalues. *SIAM Journal on Optimization*, **2**, 88–120.
[7] Grötschel, M., Lovasz, L. and Schrijver, A., 1993, *Geometric Algorithms and Combinatorial Optimization* (Berlin: Springer Verlag).
[8] Goffin, J.-L. and Vial, J.-P., 2002, Convex non-differentiable optimization: a survey focussed on the analytic center cutting plane method. *Optimization Methods and Software*, **17**, 805–867.
[9] Mitchell, J.E., 2003, Polynomial interior point cutting plane methods. *Optimization Methods and Software*, **18**, 507–534.
[10] Krishnan, K. and Mitchell, J.E., 2001, A linear programming approach to semidefinite programming problems. Technical Report, Department of Mathematical Sciences, Rensselaer Polytechnic Institute.
[11] Goldfarb, D., 2002, The simplex method for conic programming. Technical Report, CORC, Columbia University.
[12] Oskoorouchi, M. and Goffin, J.L., 2003, The analytic center cutting plane method with semidefinite cuts. *SIAM Journal on Optimization*, **13**, 1029–1153.
[13] Helmberg, C. and Rendl, F., 2000, A spectral bundle method for semidefinite programming. *SIAM Journal on Optimization*, **10**, 673–696.
[14] Krishnan, K., Pataki, G. and Zhang, Y., 2005, *A non-polyhedral active set approach to SDP*, forthcoming.
[15] Toh, K.C., Zhao, G. and Sun, J., 2002, A multiple cut analytic center cutting plane method for semidefinite feasibility problems. *SIAM Journal on Optimization*, **12**, 1126–1146.
[16] Sun, J., Toh, K.C. and Zhao, G.Y., 2002, An analytic center cutting plane method for semidefinite feasibility problems. *Mathematics of Operation Research*, **27**, 332–346.
[17] Alizadeh, F., Haeberly, J.P.A. and Overton, M.L., 1997, Complementarity and non-degeneracy in semidefinite programming. *Mathematical Programming*, **77**, 111–128.
[18] Pataki, G., 1998, On the rank of extreme matrices in semidefinite programs and the multiplicity of optimal eigenvalues. *Mathematics of Operations Research*, **23**, 339–358.
[19] Cheney, E.W. and Goldstein, A.A., 1959, Newton's method for convex programming. *Numerische Mathematik*, **1**, 253–268.
[20] Kelley, J.E., 1960, The cutting plane method for solving convex programs. *Journal of the SIAM*, **8**, 703–712.

[21] Krishnan, K., 2000, Linear Programming Approaches for Semidefinite Programming Problems. PhD thesis, Department of Mathematical Sciences, Rensselaer Polytechnic Institute.
[22] Krishnan, K. and Mitchell, J.E., 2003, Properties of a cutting plane algorithm for semidefinite programming. Technical Report, Department of Mathematical Sciences, Rensselaer Polytechnic Institute.
[23] Krishnan, K. and Terlaky, T., 2005, Interior point and semidefinite approaches in combinatorial optimization. In: D. Avis, A. Hertz, and O. Marcotte (Eds) *Graph Theory and Combinatorial Optimization* (New York: Springer), pp. 101–158.
[24] Anstreicher, K.M., 1999, Towards a practical volumetric cutting plane method for convex programming. *SIAM Journal on Optimization*, **9**, 190–206.
[25] Lemarechal, C., 1989, In: Optimization, G.L. Nemhauser, A.H.G. Rinnooy Kan, and M.J. Todd (Eds) *Non-differentiable Optimization* (Amsterdam-New York-Oxford-Tokyo: North-Holland), pp. 529–572.
[26] Kiwiel, K.C., 1985, Methods of Descent for Non-differentiable Optimization. *Lecture Notes in Mathematics, 1133* (Berlin: Springer Verlag).
[27] Hiriart-Urruty, J.B. and Lemarechal, C., 1993, *Convex analysis and Minimization Algorithms II* (Berlin Heidelberg: Springer Verlag).
[28] Makela, M.M. and Neittaanmaki, P., 1992, *Nonsmooth Optimization: Analysis and Algorithms with Applications to Optimal Control* (World Scientific Publishing Co.).
[29] Kiwiel, K.C., 1990, Proximity control in bundle methods for convex non-differentiable minimization. *Mathematical Programming*, **46**, 105–122.
[30] Oskoorouchi, M., 2002, The analytic center cutting plane method with semidefinite cuts. PhD thesis, Faculty of Management, McGill University, Montreal, Canada.
[31] Helmberg, C., 2000, Semidefinite Programming for Combinatorial Optimization. Habilitationsschrift, ZIB-Report ZR-00-34, Konrad-Zuse-Zentrum Berlin.
[32] Helmberg, C. and Kiwiel, K.C., 2002, A spectral bundle method with bounds. *Mathematical Programming*, **93**, 173–194.
[33] Helmberg, C. and Oustry, F., 2000, Bundle methods to minimize the maximum eigenvalue function. In H. Wolkowicz, R. Saigal, and L. Vandenberghe, (Eds) *Handbook of Semidefinite Programming* (Boston-Dordrecht-London: Kluwer Academic Publishers), pp. 307–337.
[34] Oustry, F., 2000, A second order bundle method to minimize the maximum eigenvalue function. *Mathematical Programming*, **89**(1), 1–33.
[35] Pataki, G., 1996, Cone-LP's and Semidefinite Programs: Geometry and a Simplex-type Method. In: W.H. Cunningham, S.T. McCormick, and M. Queyranne (Eds) *Proceedings of the Fifth IPCO Conference* (Springer Verlag).
[36] Pataki, G., 2000, The geometry of semidefinite programming. In: H. Wolkowicz, R. Saigal, and L. Vandenberghe (Eds) *Handbook of Semidefinite Programming: Theory, Algorithms, and Applications* (Kluwer Academic Publishers).
[37] Krishnan, K. and Mitchell, J.E., 2004, Semidefinite based cut and price approaches for the maxcut problem. AdvOl-Report 2004/5, Department of Computing & Software, McMaster University.
[38] Helmberg, C., 2003, Numerical evaluation of SB method. *Mathematical Programming*, **95**, 381–406.