

# Properties of a Cutting Plane Method for Semidefinite Programming<sup>1</sup>

**Kartik Krishnan Sivaramakrishnan**

Department of Mathematics  
North Carolina State University  
Raleigh, NC 27695-8205  
kksivara@ncsu.edu  
<http://www4.ncsu.edu/~kksivara>

**John E. Mitchell**

Mathematical Sciences  
Rensselaer Polytechnic Institute  
Troy, NY 12180  
mitchj@rpi.edu  
<http://www.rpi.edu/~mitchj>

Last Revised September, 2007 (submitted for publication).

## Abstract

We analyze the properties of an interior point cutting plane algorithm that is based on a semi-infinite linear formulation of the dual semidefinite program. The cutting plane algorithm approximately solves a linear relaxation of the dual semidefinite program in every iteration and relies on a separation oracle that returns linear cutting planes. We show that the complexity of a variant of the interior point cutting plane algorithm is slightly smaller than that of a direct interior point solver for semidefinite programs where the number of constraints is approximately equal to the dimension of the matrix. Our primary focus in this paper is the design of good separation oracles that return cutting planes that support the feasible region of the dual semidefinite program. Furthermore, we introduce a concept called the *tangent space induced by a supporting hyperplane* that measures the strength of a cutting plane, characterize the supporting hyperplanes that give higher dimensional tangent spaces, and show how such cutting planes can be found efficiently. Our procedures are analogous to finding facets of an integer polytope in cutting plane methods for integer programming. We illustrate these concepts with two examples in the paper. Finally, we describe separation oracles that return nonpolyhedral cutting surfaces. Recently, Krishnan et al. [41] and Oskoorouchi and Goffin [32] have adopted these separation oracles in conic interior point cutting plane algorithms for solving semidefinite programs.

**Keywords:** Semidefinite programming, interior point methods, regularized cutting plane algorithms, maximum eigenvalue function, cone of tangents.

---

<sup>1</sup>The first author is supported in part by a startup grant from NC State University. The second author is supported in part by NSF grant numbers DMS-0317323 and DMS-0715446.

# 1 Introduction

A semidefinite programming problem requires minimizing a linear objective function in symmetric matrix variables subject to linear equality constraints together with a convex constraint that these variables be positive semidefinite. The tremendous activity in semidefinite programming was spurred by the discovery of efficient interior point algorithms (see de Klerk [10]) for solving semidefinite programs; and important applications of semidefinite programming in combinatorial optimization, control, robust optimization, and more recently in polynomial optimization (see Ben-Tal and Nemirovskii [5] and Laurent and Rendl [24]).

Primal-dual interior point methods (IPMs) (see de Klerk [10] and Monteiro [27]) are currently the most popular techniques for solving semidefinite programs. However, current semidefinite solvers based on IPMs can only handle problems with dimension  $n$  and number of equality constraints  $k$  up to a few thousands. Each iteration of a primal-dual IPM solver needs to form a dense Schur matrix, store this matrix in memory, and finally factorize and solve a dense system of linear equations of size  $k$  with this coefficient matrix. Several techniques have recently been developed to solve large scale SDPs; these include: the low rank factorization approach of Burer & Monteiro [8]; the spectral bundle methods of Helmberg et al. [13, 14, 15]) and Nayakkankuppam [28]; parallel implementations of primal-dual IPMs on shared memory (see Borchers and Young [7]) and distributed memory (see Yamashita et al. [44]) systems; and interior point cutting plane algorithms (see Krishnan et al. [40, 41], Oskoorouchi et al. [32, 33], and Sherali and Fraticelli [39]).

In this paper, we investigate the properties of the interior point cutting algorithms presented in Krishnan et al. [20, 21, 41]. The methods are based on a semi-infinite linear formulation of a semidefinite program and they use an interior point cutting framework to approximately solve the underlying semidefinite program to 2-3 digits of accuracy. We introduce the semi-infinite formulation in Section 2 and give a brief description of the algorithm in section 3. The algorithm uses a primal-dual interior point method to approximately solve the linear relaxations that arise at each iteration of the cutting plane algorithm. Theoretically, it is possible to use a volumetric interior point method to solve the relaxations. We show in Section 4 that the complexity of a volumetric interior point cutting plane algorithm for solving a semidefinite program to a prescribed tolerance  $\epsilon > 0$  is slightly less than that of a primal-dual IPM, at least for a semidefinite program where the number of constraints is approximately equal to the dimension of the matrix.

Our interior cutting plane algorithm solves a linear relaxation of the dual semidefinite program. In every iteration, the algorithm calls a separation oracle that adds cutting planes to strengthen the current relaxation. The convergence of the cutting plane algorithm can be improved by adding strong cutting planes. We show that we can always find a hyperplane that supports the feasible region of the semidefinite program. In cutting plane methods

for integer programming (see Nemhauser and Wolsey [29]), the strongest cutting planes are facet inequalities that describe the higher dimensional faces of the integer polytope. Each supporting hyperplane induces a corresponding tangent space (we define the notion in Section 5). The dimension of the tangent space measures the strength of this hyperplane. Suppose, the primal semidefinite program has  $k$  constraints and the nullity of the dual slack matrix at the support point is 1, we show that the dimension of the tangent space defined by the hyperplane is  $k - 1$ . Further, if the nullity of the active dual slack matrix is  $r$ , we describe how to generate a cutting plane that gives a tangent space of dimension at least  $k - r$ . We illustrate these concepts with two representative examples in Section 6. The strength of nonpolyhedral cutting surfaces is the subject of Section 7. These cutting surfaces have been incorporated in an overall conic interior point decomposition algorithm for semidefinite programming — see Krishnan et al. [41] for details and computational experiences with such an algorithm.

**Notation:** The set of  $n \times n$  symmetric matrices is denoted  $\mathcal{S}^n$ . The set of positive semidefinite  $n \times n$  matrices is denoted  $\mathcal{S}_+^n$ . The requirement that a matrix be positive semidefinite is written  $X \succeq 0$ . Matrices are represented using upper case letters and vectors using lower case letters. Given an  $n$ -vector  $v$ , the diagonal  $n \times n$  matrix with the  $i$ th diagonal entry equal to  $v_i$  for  $i = 1, \dots, n$  is denoted  $\text{Diag}(v)$ . The  $n \times n$  identity matrix is denoted  $I_n$ ; when the dimension is clear from the context, we omit the subscript. The Frobenius inner product of two  $m \times n$  matrices  $A$  and  $B$  is denoted  $A \bullet B$ ; if the matrices are symmetric this is equal to the trace of their product, denoted  $\text{trace}(AB)$ .

## 2 Semi-infinite formulations for semidefinite programming

Consider the semidefinite programming problem

$$\begin{aligned} \min \quad & C \bullet X \\ \text{s.t.} \quad & \mathcal{A}(X) = b \quad (\text{SDP}) \\ & X \succeq 0 \end{aligned}$$

with dual

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

where  $X, S \in \mathcal{S}_+^n$ ,  $C \in \mathcal{S}^n$ ,  $b$  and  $y$  are vectors in  $\mathbb{R}^k$ , and  $\mathcal{A}$  is a linear function mapping  $\mathcal{S}^n$  to  $\mathbb{R}^k$ . We can regard  $\mathcal{A}$  as being composed of  $k$  linear functions, each represented by a matrix  $A_i \in \mathcal{S}^n$ , so the constraint  $\mathcal{A}(X) = b$  is equivalent to the  $k$  linear constraints  $A_i \bullet X = b_i$ ,  $i = 1, \dots, k$ . The expression  $\mathcal{A}^T y$  is equivalent to  $\sum_{i=1}^k y_i A_i$ .

We make the following two assumptions.

**Assumption 1** *The matrices  $A_i$ ,  $i = 1, \dots, k$  are linearly independent in  $\mathcal{S}^n$ .*

**Assumption 2** *There exists a constant  $a \geq 0$  such that every  $X$  satisfying  $\mathcal{A}X = b$  also satisfies  $\text{trace}(X) = a$ .*

Helmberg [12] shows that every semidefinite program whose primal feasible set is bounded can be rewritten to satisfy Assumption 2. The following lemma is a consequence of this assumption.

**Lemma 1** *There exists a unique vector  $\hat{y}$  satisfying  $\mathcal{A}^T \hat{y} = I$ , the identity matrix.*

**Proof:** The constraint  $\text{trace}(X) = I \bullet X = a$  is implied by the  $k$  constraints  $A_i \bullet X = b_i$ . Therefore,  $I$  must be a linear combination of the matrices  $A_i$ , showing the existence of  $\hat{y}$ . The uniqueness of  $\hat{y}$  follows from Assumption 1, the linear independence of the matrices  $A_i$ .  $\square$

Consider any feasible point  $y$  in (SDD). The point  $(y - \lambda \hat{y})$  is strictly feasible in (SDD) for  $\lambda > 0$ . Indeed, the dual slack matrix at this new point is  $S = (C - \mathcal{A}^T(y - \lambda \hat{y})) = (C - \mathcal{A}^T y + \lambda I) \succ 0$ . So, Assumption 2 ensures that (SDD) has a strictly feasible (Slater) point. This assumption ensures that we have *strong duality* at optimality, i.e. the optimal objective values of (SDP) and (SDD) are equal. Moreover, the primal problem (SDP) attains its optimal solution.

Note that the convex constraint  $X \succeq 0$  is equivalent to

$$\eta^T X \eta = \eta \eta^T \bullet X \geq 0 \quad \forall \eta \in B \quad (1)$$

where  $B$  is a compact set, typically  $\{\eta : \|\eta\|_2 \leq 1\}$  or  $\{\eta : \|\eta\|_\infty \leq 1\}$  that are commonly used in *trust region methods* (Conn et al. [9]). These constraints are linear inequalities in the matrix variable  $X$ , but there is an *infinite* number of them. Thus, a semidefinite program is a *semi-infinite* linear program in  $\mathbb{R}^{\frac{n(n+1)}{2}}$  variables. The survey paper by Hettich and Kortanek [16] discusses theory, algorithms, and applications of semi-infinite programming.

We now consider two semi-infinite linear programs (PSIP)

$$\begin{aligned} \min \quad & C \bullet X \\ \text{s.t.} \quad & \mathcal{A}(X) = b \\ & \eta^T X \eta \geq 0, \quad \forall \eta \in B \end{aligned} \quad (\text{PSIP})$$

$$\begin{aligned} \max \quad & b^T y \\ \text{s.t.} \quad & \mathcal{A}^T y + S = C \\ & \eta^T S \eta \geq 0, \quad \forall \eta \in B \end{aligned} \quad (\text{DSIP})$$

for (SDP) and (SDD), respectively. Note that  $X$  is  $n \times n$  and symmetric, so (PSIP) has  $\binom{n+1}{2} = \frac{n(n+1)}{2} = O(n^2)$  variables. In contrast, there are  $k$  variables in (DSIP). We have  $k \leq \binom{n+1}{2}$  (from Assumption 1). Therefore, it is more efficient to deal with (DSIP), since

we are dealing with smaller linear programs (but see also the discussion at the end of this section).

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

$$\begin{aligned} \max \quad & b^T y \\ \text{s.t.} \quad & \sum_{j=1}^k y_j (\eta_i^T A_j \eta_i) \leq \eta_i^T C \eta_i, \quad i = 1, \dots, m \quad (\text{LDR}) \end{aligned}$$

of (SDD). The dual to (LDR) can be expressed as follows:

$$\begin{aligned} \min \quad & C \bullet \left( \sum_{i=1}^m x_i \eta_i \eta_i^T \right) \\ \text{s.t.} \quad & \mathcal{A} \left( \sum_{i=1}^m x_i \eta_i \eta_i^T \right) = b \quad (\text{LPR}) \\ & x_i \geq 0, \quad i = 1, \dots, m. \end{aligned}$$

The convex constraint  $S \succeq 0$  is also equivalent to

$$P^T S P \succeq 0, \quad \forall P \in \mathbb{R}^{n \times r}, \quad r < n, \quad \text{and} \quad P^T P = I_r. \quad (2)$$

This allows one to develop a semi-infinite semidefinite formulation for (SDD) where the semidefinite cone of size  $n$  in (SDD) is replaced with an infinite number of semidefinite cones of size  $r < n$ . An overview of various techniques to update relaxations involving finite subsets of these cones in interior cutting plane algorithms can be found in Krishnan and Mitchell [22]. In particular, in Krishnan and Mitchell [21] the relaxations are linear programs; in Krishnan et al. [41] and Oskoorouchi and Goffin [32], the relaxations are conic programs over a linear cone and several semidefinite cones of small size; and in the spectral bundle algorithm of Helmberg and Rendl [15], the relaxations are conic programs over one linear cone and one semidefinite cone.

**Theorem 1** *Let  $y^*$  and  $x^*$  be optimal solutions to (LDR) and (LPR), respectively.*

1. *The matrix  $X^* = \sum_{i=1}^m x_i^* \eta_i \eta_i^T$  is feasible in (SDP).*
2. *Let  $S^* = (C - \mathcal{A}^T y^*)$ . We have  $X^* \bullet S^* = 0$ . Furthermore, if  $S^* \succeq 0$  then  $X^* S^* = 0$ , and  $X^*$  is an optimal solution to (SDP).*

**Proof:** The primal conic program (LPR) is a constrained version of (SDP). Therefore, any feasible solution  $x^*$  in (LDR) gives a  $X = \sum_{i=1}^m x_i^* \eta_i \eta_i^T$  that is feasible in (SDP). We have

$$\begin{aligned} X^* \bullet S^* &= \left( \sum_{i=1}^m x_i^* \eta_i \eta_i^T \right) \bullet (C - \mathcal{A}^T y^*) \\ &= \sum_{i=1}^m x_i^* \eta_i^T (C - \mathcal{A}^T y^*) \eta_i \\ &= 0 \end{aligned}$$

from the complementary slackness at optimality for (LPR) and (LDR). If  $S^* = (C - \mathcal{A}^T y^*)$  is positive semidefinite, then it is feasible in (SDD). Moreover,  $X^* \succeq 0$ ,  $S^* \succeq 0$ , and  $X^* \bullet S^* = 0$  together imply  $X^* S^* = 0$  (see Alizadeh et al. [1]).  $\square$

The following theorem is due to Pataki [35] (also see Alizadeh et al. [1]):

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

The theorem suggests that an upper bound on the rank of an optimal solution  $X^*$  is  $r^* = \lfloor \sqrt{2k} \rfloor$ , where  $k$  is the number of equality constraints in (SDP). Suppose  $S^*$  is the dual slack matrix at an optimal solution  $y^*$  to (SDD). The complementary slackness conditions  $X^* S^* = 0$  at optimality suggest that  $X^*$  and  $S^*$  share a common eigenspace. Moreover, the positive eigenspace of  $X^*$  corresponds to the null space of  $S^*$  (see Alizadeh et al. [1]). Therefore, Theorem 2 also provides an upper bound on the dimension of the null space of  $S^*$ . We will utilize this observation later in sections 5.3 and 7 in this paper.

Let  $q = \frac{n(n+1)}{2} - k$ . It is possible to use a nullspace representation to reformulate (SDP) as a semi-infinite programming problem with  $q$  variables. This is advantageous if  $q$  is smaller than  $k$ , in particular if  $q$  is  $O(n)$ . Let  $\mathcal{B} : \mathcal{S}^n \rightarrow \mathbb{R}^q$  be the null space operator corresponding to  $\mathcal{A}$ , so the range of  $\mathcal{B}^T$  is exactly the kernel 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 the linear equality constraints  $\mathcal{A}(X) = b$ . The set of feasible solutions to  $\mathcal{A}(X) = b$  is the set of all matrices of the form  $X = X^0 - \mathcal{B}^T(u)$  for some  $u \in \mathbb{R}^q$ . The problem (SDP) can then be written equivalently as

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

The problem (SDPN) is in exactly the form of (SDD), so we can construct a linear programming relaxation of it in the form (LDR), with  $q$  variables. We return to this alternative representation when discussing the complexity of the algorithm in section 4. (A similar nullspace representation of linear programming problems has been analyzed in the interior point literature; see, for example, Todd and Ye [42] and Zhang et al. [45].)

### 3 Cutting plane algorithms for semidefinite programming

Let

$$\begin{aligned} Y &= \{y \in \mathbb{R}^k : S = (C - \mathcal{A}^T y) \succeq 0\} \\ &= \{y \in \mathbb{R}^k : \lambda_{\max}(A^T y - C) \leq 0\} \end{aligned} \tag{3}$$

be the convex set containing the feasible solutions to (SDD). The goal of cutting plane methods that solve (SDD) is to find an optimal point that maximizes  $b^T y$  over  $Y$ . There are three important ingredients in this algorithm:

1. The technique used to update the relaxations (LPR) and (LDR) in every iteration.
2. The choice of the query point  $\bar{y}$ .
3. Given a query point  $\bar{y}$ , a *separation oracle* that either (a) tells us that  $\bar{y} \in Y$  in which case we try to improve the objective function  $b^T y$ , or (b) returns a separating hyperplane that separates  $\bar{y}$  from the set  $Y$ .

Choosing an optimal solution  $\bar{y}$  to (LDR) as the next query point is a bad idea (see Example 1.1.2 on page 277 of Hiriart-Urruty and Lemaréchal [18]). Instead, one adopts strategies where the idea is to choose a more central point  $\bar{y}$  in the feasible region of (LDR) as the next query point. These strategies include using a quadratic penalty term as in a bundle method (see Helmberg and Rendl [15]), using an analytic center with objective function cuts used to push the iterates towards an optimal solution (see Oskoorouchi and Goffin [32], for example), and only solving (LDR) approximately with an interior point method (see Krishnan et al. [41], for example).

The main contribution of this paper is to design an efficient separation oracle that can be utilized within a cutting plane algorithm for solving (SDD). We discuss the choice of good cutting planes in section 5 and illustrate these choices with examples in Section 6. We later generalize the separation oracle to return a nonpolyhedral separating hypersurface in section 7.

We first introduce several separating hyperplanes for the feasible set  $Y$ . The maximum eigenvalue function  $\lambda_{\max}(\mathcal{A}^T y - C)$  is a convex non-smooth function, with a discontinuous gradient whenever this eigenvalue has a multiplicity greater than one. The subdifferential to  $\lambda_{\max}(\mathcal{A}^T y - C)$  at  $y = \bar{y}$  (see Hiriart-Urruty and Lemaréchal [18]) is given by

$$\partial \lambda_{\max}(\mathcal{A}^T \bar{y} - C) = \text{conv}\{\mathcal{A}(pp^T) : p^T(\mathcal{A}^T \bar{y} - C)p = \lambda_{\max}(\mathcal{A}^T \bar{y} - C), p^T p = 1\} \quad (4)$$

where  $\text{conv}$  denotes the convex hull operation. There is also an alternate description

$$\partial \lambda_{\max}(\mathcal{A}^T \bar{y} - C) = \{\mathcal{A}(PVP^T) : \text{trace}(V) = 1, V \succeq 0\} \quad (5)$$

where  $P \in \mathbb{R}^{n \times r}$  is an orthonormal matrix containing the eigenspace of  $\lambda_{\max}(\mathcal{A}^T \bar{y} - C)$  (see Overton [34]). We note that (5) does not involve the convex hull operation, and this description will be used later in sections 5.1 and 7 in this paper. Any subgradient from (4) gives the valid inequality

$$\lambda_{\max}(\mathcal{A}^T y - C) \geq \lambda_{\max}(\mathcal{A}^T \bar{y} - C) + \mathcal{A}(pp^T)^T(y - \bar{y}) \quad \forall y. \quad (6)$$

Now given the query point  $\bar{y}$ , we first check for feasibility, i.e.  $\lambda_{\max}(\mathcal{A}^T \bar{y} - C) \leq 0$ . If  $\bar{y}$  is not feasible, then we can construct a cut

$$\lambda_{\max}(\mathcal{A}^T \bar{y} - C) + \mathcal{A}(pp^T)^T(y - \bar{y}) \leq 0 \quad (7)$$

To motivate this consider (7) with the reversed inequality. This would imply  $\lambda_{\max}(\mathcal{A}^T y - C) > 0$  from (6), and so  $y$  also violates the convex constraint. It follows that any feasible  $y$  satisfies (7). Using the fact that  $p$  is a normalized eigenvector corresponding to  $\lambda_{\max}(\mathcal{A}^T \bar{y} - C)$ , we can rewrite (7) as

$$p^T(C - \mathcal{A}^T y)p \geq 0 \quad (8)$$

which is a valid cutting plane which is satisfied by all the feasible  $y$ . From linear algebra (see Horn and Johnson [19]), we have

$$\lambda_{\max}(\mathcal{A}^T \bar{y} - C) = \max \{ \eta^T (\mathcal{A}^T \bar{y} - C) \eta : \|\eta\|_2 = 1 \}. \quad (9)$$

Moreover, any eigenvector  $\eta$  corresponding to positive eigenvalue of the matrix  $(\mathcal{A}^T \bar{y} - C)$  gives a valid cutting plane  $\eta^T(C - \mathcal{A}^T y)\eta \geq 0$ . However, these cutting planes are weaker than the cutting plane (8) that corresponds to the most positive eigenvalue of  $(\mathcal{A}^T \bar{y} - C)$ . Cutting planes can also be found using the  $\infty$ -norm; for details, see Krishnan and Mitchell [23].

Algorithm 1 presents the overall interior point cutting plane algorithm for solving semidefinite programs. For the implementation details of a more sophisticated algorithm similar to Algorithm 1 see Krishnan et al. [41]. Procedures to warm-start the new relaxations with strictly feasible starting points are discussed in Krishnan et al. [41]. These are extensions of techniques given in Mitchell and Todd [26]. The relaxations are only solved approximately in Step 3 because the iterates are then more centered, which leads to stronger cutting planes and improved warm starting. We show later in theorem 4 that the vector  $\check{y} = y^m - \lambda \hat{y}$  constructed in Step 6 is on the boundary of  $Y$  and provides a lower bound. The algorithm can be refined to drop unimportant constraints; for details see Krishnan et al. [41]. We illustrate two iterations of the cutting plane algorithm in figure 1.

Computational results with Algorithm 1 can be found in Krishnan et al. [41]. Moreover, computational results obtained with the related spectral bundle and ACCPM algorithms can be found in Helmberg et al. [13, 14, 15] and Oskoorouchi et al. [32, 33], respectively.

### Algorithm 1 (Interior point cutting plane algorithm)

1. **Initialize:** Set  $m = 1$ . Choose an initial LP relaxation (LDR) that has a bounded optimal solution. Let  $x^m$  and  $y^m$  be feasible solutions in (LPR) and (LDR), respectively. Choose a tolerance parameter  $\beta > 0$  and a termination parameter  $\epsilon > 0$ .
2. **Warm start the current relaxation:** Generate strictly feasible starting points for (LPR) and (LDR).
3. **Solve the current relaxation:** Solve the current relaxations (LPR) and (LDR) with the strictly feasible starting points from Step 2 to a tolerance  $\beta$ . Let  $x^m$  and  $y^m$  be the current solutions to (LPR) and (LDR), respectively. Update the upper bound using the objective value to (LPR).
4. **Separation Oracle:** Call the separation oracle at the point  $y^m$ . If the oracle returns a cutting plane, update (LPR) and (LDR).
5. **Optimality check:** If the oracle reported feasibility in Step 4 then reduce  $\beta$  by a constant factor. Else,  $\beta$  is unchanged. If  $\beta < \epsilon$ , we have an optimal solution, STOP.
6. **Update lower bound:** If the oracle reported feasibility in Step 4 then  $b^T y^m$  provides the current lower bound on the optimal objective value. Else,  $S^m = (C - A^T y^m)$  is not psd. Perturb  $S^m$  (using the vector  $\hat{y}$  from Lemma 2) to generate  $\tilde{y}$ , that is on the boundary of  $Y$  and whose objective value  $b^T \tilde{y}$  provides a lower bound. Update the lower bound.
7. **Loop:** Set  $m = m + 1$  and return to Step 2.

## 4 Complexity of the interior point cutting plane algorithm

It must be mentioned that a semidefinite program was known to be solvable in polynomial time, much before the advent of interior point methods. In fact we can use the polynomial time oracle for a semidefinite program mentioned in section 3 in conjunction with the ellipsoid algorithm to solve this problem in a polynomial number of arithmetic operations. It is interesting to compare the worst case complexity of such a method, with that of interior point methods.

The ellipsoid algorithm (for example, see Grötschel et al. [11]) can solve a convex programming problem of size  $k$  with a separation oracle to an accuracy of  $\epsilon$ , in  $O(k^2 \log(\frac{1}{\epsilon}))$

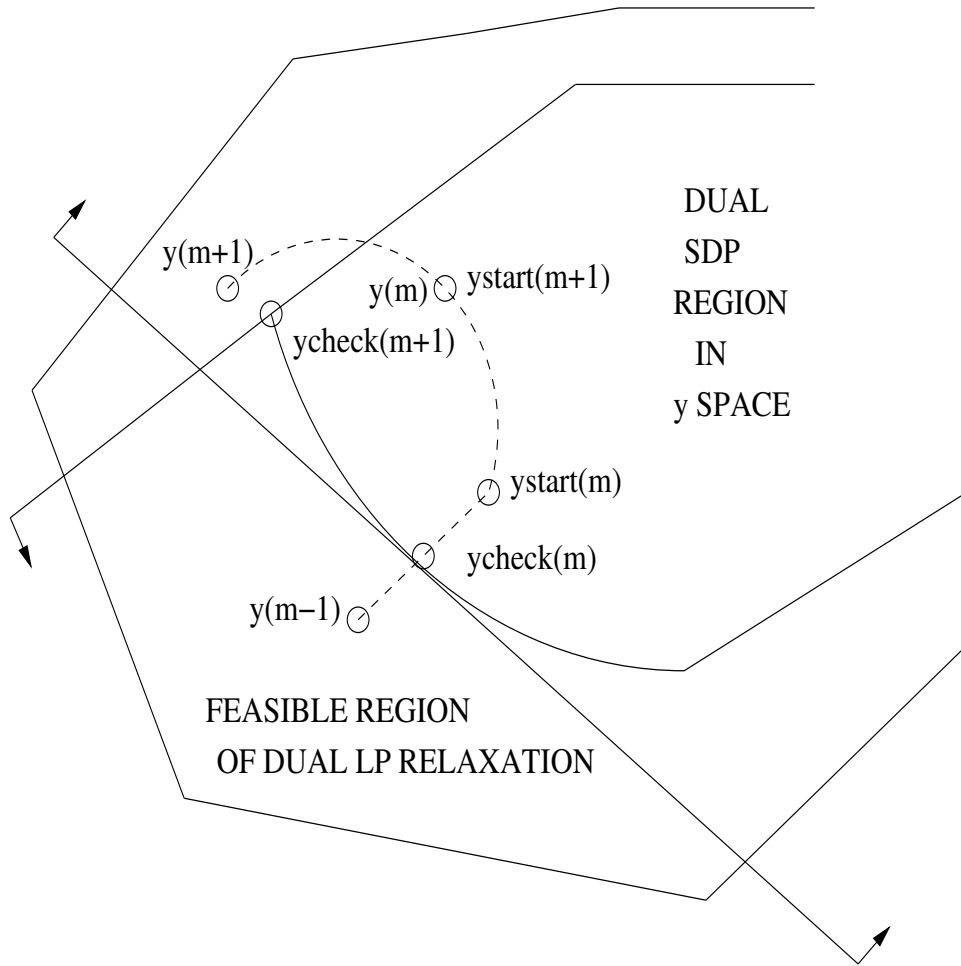


Figure 1: Solving the SDP via a cutting plane algorithm: In the  $(m-1)$ th iteration, the point  $y(m-1)$  is outside the feasible region of (SDD). The oracle returns the unique hyperplane that supports the dual SDP feasible region at  $ycheck(m)$  and cuts off  $y(m-1)$ . A strictly feasible restart point  $ystart(m)$  is found. The new LP relaxation is approximately solved using an interior point method, giving  $y(m)$ , that is inside the feasible region of (SDD). In this case, we tighten the tolerance to which the LP relaxation is solved, while the feasible region of the relaxation is unchanged. The LP is solved with starting point  $y(m)$ , giving  $y(m+1)$  that is outside the feasible region of (SDD). The oracle returns a cutting plane that supports the dual feasible region at  $ycheck(m+1)$ . Note that the boundary of (SDD) has a kink at  $ycheck(m+1)$ , and there are several supporting hyperplanes at this point. The separation oracle returns the tightest hyperplane at  $ycheck(m+1)$  that cuts off  $y(m+1)$ . Note that this hyperplane corresponds to a higher dimensional face of SDP feasible region.

calls to this oracle and in  $O(k^2 \log(\frac{1}{\epsilon})T + k^4 \log(\frac{1}{\epsilon}))$  arithmetic operations, where  $T$  is the number of operations required for one call to the oracle. Each iteration of the ellipsoid algorithm requires  $O(k^2)$  arithmetic operations.

The interior point cutting plane algorithm with the best complexity is a volumetric barrier method, due to Vaidya [43] and refined by Anstreicher [2, 3] and Ramaswamy and Mitchell [38]. The volumetric center minimizes the determinant of the Hessian of the standard potential function  $-\sum_i \ln s_i$ , where  $s$  is the vector of dual slacks in the linear programming relaxation. This is closely related to finding the point where the volume of the inscribing Dikin ellipsoid is largest. (See Mitchell [25] for a survey of interior point polynomial time cutting plane algorithms). This algorithm requires  $O(k \log(\frac{1}{\epsilon}))$  iterations, with each iteration requiring one call to the oracle and  $O(k^3)$  other arithmetic operations. Thus, the overall complexity is  $O(k \log(\frac{1}{\epsilon})T + k^4 \log(\frac{1}{\epsilon}))$  arithmetic operations. Note that the number of calls to the oracle required by the volumetric algorithm is smaller than the corresponding number for the ellipsoid algorithm. This complexity of  $O(k \log(\frac{1}{\epsilon}))$  calls to the separation oracle is optimal — see Nemirovskii and Yudin [30].

The oracle for semidefinite programming requires the determination of an eigenvector corresponding to the smallest eigenvalue of the current dual slack matrix. Let us examine the arithmetic complexity of this oracle. Let us assume that our current iterate is  $\bar{y} \in \mathbb{R}^k$ .

1. We first have to compute the dual slack matrix  $\bar{S} = (C - \sum_{i=1}^k \bar{y}_i A_i)$ , where  $C, \bar{S}$ , and  $A_i, i = 1, \dots, k$  are in  $S^n$ . This can be done in  $O(kn^2)$  arithmetic operations.
2. We then compute  $\lambda_{\min}(\bar{S})$ , and an associated eigenvector  $\eta$ . This can be done in  $O(n^3)$  arithmetic operations using the QR algorithm for computing eigenvalues, and possibly in  $O(n^2)$  operations using the Lanczos scheme, whenever  $S$  is sparse.
3. If  $\lambda_{\min}(\bar{S}) \geq 0$ , we are feasible, and therefore we cut based on the objective function. This involves computing the gradient of the linear function, and this can be done in  $O(k)$  time.
4. On the other hand if  $\lambda_{\min}(\bar{S}) < 0$ , then we are yet outside the SDP cone; we can now add the valid constraint  $\sum_{i=1}^k y_i (\eta^T A_i \eta) \leq \eta^T C \eta$ , which cuts off the current infeasible iterate  $\bar{y}$ . The coefficients of this constraint can be computed in  $O(kn^2)$  arithmetic operations.

It follows that the entire oracle can be implemented in  $T = O(n^3 + kn^2)$  time.

We summarize this discussion in the following theorem.

**Theorem 3** *A volumetric cutting plane algorithm for a semidefinite programming problem of size  $n$  with  $k$  constraints requires  $O((kn^3 + k^2n^2 + k^4) \log(\frac{1}{\epsilon}))$  arithmetic operations. An*

*ellipsoid algorithm cutting plane method requires  $O((k^2n^3 + k^3n^2 + k^4) \log(\frac{1}{\epsilon}))$  arithmetic operations.*

Let us compare this with a direct interior point approach. Interior point methods (see Monteiro [27] for more details) can solve an SDP of size  $n$ , to a precision  $\epsilon$ , in  $O(\sqrt{n} \log(\frac{1}{\epsilon}))$  iterations (this analysis is for a short step algorithm). As regards the complexity of an iteration :

1. We need  $O(kn^3 + k^2n^2)$  arithmetic operations to form the Schur matrix  $M$ . This can be brought down to  $O(kn^2 + k^2n)$  if the constraint matrices  $A_i$  are rank one as in the semidefinite relaxation of the maxcut problem (see Laurent and Rendl [24]).
2. We need  $O(k^3)$  arithmetic operations to factorize the Schur matrix, and compute the search direction. Again, this number can be brought down if we employ iterative methods.

The overall scheme can thus be carried out in  $O(k(n^3 + kn^2 + k^2)\sqrt{n} \log(\frac{1}{\epsilon}))$  arithmetic operations. (We may be able to use some partial updating strategies to factorize  $M$  and improve on this complexity). Thus, if  $k = O(n)$  then the complexity of the volumetric cutting plane algorithm is slightly smaller than that of the direct primal-dual interior point method. Thus we could in theory improve the complexity of solving an SDP using a cutting plane approach.

Note also that if  $q = \frac{n(n+1)}{2} - k$  is  $O(n)$ , we can use the nullspace representation (SPDN) to improve the complexity estimate given in theorem 3. In particular, the problem (SDPN) is in exactly the form of (SDD), so the cutting plane approach of section 3 can be applied to it directly. It follows from theorem 3 that (SDP) can be solved in  $O((qn^3 + q^2n^2 + q^4) \log(\frac{1}{\epsilon}))$  arithmetic operations using a volumetric barrier cutting plane algorithm. This is again superior to the complexity derived above for a direct interior point method for solving (SDP) if  $q = O(n)$ .

Anstreicher [4] has considered a volumetric barrier algorithm for semidefinite programming, strengthening results in Nesterov and Nemirovskii [31]. Like primal-dual interior point methods, this volumetric barrier algorithm works explicitly with the full formulation (SDP) and (SDD). Given a dual feasible solution  $y$  for (SDD), with corresponding dual slack matrix  $S$ , let  $f(y) = -\ln \det(S)$ . The volumetric barrier function is then  $V(y) := \frac{1}{2} \ln \det(\nabla^2 f(y))$ . The number of iterations required is similar to that for a primal-dual interior point method if  $k$  and  $n$  are similar in size.

## 5 Properties and generation of good cutting planes

We can improve the convergence of the interior point cutting plane algorithm by developing a better separation oracle. We first investigate the valid inequalities returned by the

separation oracle in the cutting plane algorithm.

Given a point  $\bar{y} \notin Y$ , the cutting plane algorithm with the 2 norm oracle generates constraints of the form  $\eta^T(\mathcal{A}^T y)\eta \leq \eta^T C \eta$ , where  $\eta$  is an eigenvector of the current dual slack matrix  $\bar{S} = (C - \mathcal{A}^T \bar{y})$  with a negative eigenvalue. We show first that there is a point  $\check{y}$  on the boundary  $Y$  that satisfies the cutting plane at equality, if the plane is chosen appropriately.

Let  $\eta$  be an eigenvector of  $\bar{S}$  of norm one corresponding to the most negative eigenvalue of  $\bar{S}$ , and let  $\lambda$  denote this eigenvalue. Define

$$\check{y} = \bar{y} - \lambda \hat{y} \tag{10}$$

where  $\hat{y}$  is the vector in Lemma 1. The dual slack matrix at  $\check{y}$  is  $\check{S} = \bar{S} + \lambda I$ , which is positive semidefinite, so  $\check{y} \in Y$ . Further,  $\eta$  is in the nullspace of  $\check{S}$ .

**Theorem 4** *The constraint  $\eta^T(C - \mathcal{A}^T y)\eta \geq 0$  is satisfied at equality by  $\check{y}$ .*

**Proof:** We have  $\check{S} = C - \mathcal{A}^T \check{y}$  and  $\eta$  is in the nullspace of this matrix. Thus we have  $\eta^T \check{S} \eta = 0$  and so this feasible  $\check{y} \in Y$  satisfies the new constraint at equality.  $\square$

We let  $r$  denote the nullity of  $\check{S}$ . Let  $\check{S}$  have the eigendecomposition

$$\check{S} = [Q_1 \ Q_2] \begin{bmatrix} \Lambda & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} Q_1^T \\ Q_2^T \end{bmatrix}, \tag{11}$$

where  $Q = [Q_1 \ Q_2]$  is an orthonormal matrix and  $\Lambda$  is a  $(n - r) \times (n - r)$  positive diagonal matrix.

## 5.1 Cone of tangents and the normal cone

A valid inequality for a convex set gives a face of the convex set, namely the intersection of the set with the hyperplane defined by the inequality. For a full-dimensional polyhedron, the only inequalities that are necessary are those describing the facets of the polyhedron (see Nemhauser and Wolsey [29]). For a more general convex set, it is useful to consider the intersection of the cone of tangents (translated appropriately) with the supporting hyperplane. We call this intersection the tangent space defined by the hyperplane, and we define it formally in the following definition.

**Definition 1** *Let  $Y$  be a nonempty closed convex set in  $\mathbb{R}^k$ . For any point  $\bar{y} \in \mathbb{R}^k$ , the distance from  $\bar{y}$  to  $Y$  is defined as the distance from  $\bar{y}$  to the unique closest point in  $Y$  and is denoted  $\text{dist}(\bar{y}, Y)$ . Let  $\check{y}$  be a point on the boundary of  $Y$ . The cone of feasible directions, the tangent cone, and the normal cone at  $\check{y}$  are defined as*

$$\text{dir}(\check{y}, Y) = \{d : \check{y} + td \in Y \text{ for some } t > 0\}$$

$$\begin{aligned}
\text{tcone}(\check{y}, Y) &= \text{cl}(\text{dir}(\check{y}, Y)) \quad (\text{closure of } \text{dir}(\check{y}, Y)) \\
&= \{d : \text{dist}(\check{y} + td, Y) = O(t^2)\} \\
\text{ncone}(\check{y}, Y) &= \{v : d^T v \leq 0, \forall d \in \text{tcone}(\check{y}, Y)\}.
\end{aligned}$$

Given a supporting hyperplane  $H$  for  $Y$ , the tangent space defined by  $H$  at  $\check{y}$  is

$$\text{tpl}(\check{y}, Y, H) = \{d \in \text{tcone}(\check{y}, Y) : \check{y} + d \in H\}.$$

A convex subset  $F$  of  $Y$  is a face of  $Y$  if

$$x \in F, y, z \in Y, x \in (y, z) \text{ implies } y, z \in F,$$

where  $(y, z)$  denotes the line segment joining  $y$  and  $z$ . Note that  $\{u : u = \check{y} + d, d \in \text{tpl}(\check{y}, Y, H)\}$  contains the face  $H \cap Y$  of  $Y$ .

The equivalence of the two definitions of  $\text{tcone}$  follows from page 135 of Hiriart-Urruty and Lemaréchal [17]. Note that our definition of the tangent space defined by a hyperplane is a one-sided definition, that is, it depends on  $d \in \text{tcone}(\check{y}, Y)$ . Conceptually, the idea of a tangent space defined by a hyperplane will be used as an analogue of the idea of a face of a polyhedron. The geometry of semidefinite programming is surveyed by Pataki [36].

We show that if  $Y$  is full-dimensional and if the inequality arises from an eigenvector of  $S$  then the dimension of the tangent space is related to the dimension of the corresponding eigenspace.

We will now derive an expression for  $\text{ncone}(\check{y}, Y)$ .

**Theorem 5** Assume that  $\check{y}$  satisfies  $\lambda_{\max}(\mathcal{A}^T \check{y} - C) = 0$  and there exists a Slater point  $y_s$  such that  $\lambda_{\max}(\mathcal{A}^T y_s - C) < 0$ . Then

$$\begin{aligned}
\text{ncone}(\check{y}, Y) &= \text{cone}(\partial \lambda_{\max}(\mathcal{A}^T \check{y} - C)) \\
&= \{\mathcal{A}(Q_2 V Q_2^T) : V \succeq 0\}
\end{aligned} \tag{12}$$

where  $\text{cone}(X) = \{\gamma x : x \in X, \gamma \geq 0\}$  and  $Q_2$  is the nullspace of  $(\mathcal{A}^T \check{y} - C)$ .

**Proof:** The proof for the first expression for  $\text{ncone}(\check{y}, Y)$  can be found in theorem 1.3.5 on page 245 of Hiriart-Urruty and Lemaréchal [17]. Using the second expression (5) for the subdifferential  $\partial \lambda_{\max}(\mathcal{A}^T \check{y} - C)$ , we have

$$\begin{aligned}
\text{ncone}(\check{y}, Y) &= \{\gamma \mathcal{A}(Q_2 V Q_2^T) : \text{trace}(V) = 1, V \succeq 0, \gamma \geq 0\} \\
&= \{\mathcal{A}(Q_2 V Q_2^T) : V \succeq 0\}.
\end{aligned}$$

□

It should be noted that the cone  $\{\mathcal{A}(Q_2 V Q_2^T) : V \succeq 0\}$  may not be closed. Consider the following example drawn from Pataki [37]:

**Example 1** Take  $k = n = 2$ . Let

$$Q_2^T A_1 Q_2 = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \quad Q_2^T A_2 Q_2 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}.$$

It can be easily verified that the ray  $[0 \ 1]^T$  is not in the cone  $\{\mathcal{A}(Q_2 V Q_2^T) : V \succeq 0\}$ , but it is in its closure. It should be noted that the matrices  $A_1$  and  $A_2$  in this example do not satisfy Assumption 2.

A sufficient condition (see Pataki [37]) to guarantee the closure of the cone  $\{\mathcal{A}(Q_2 V Q_2^T) : V \succeq 0\}$  is the existence of a  $y \in \mathbb{R}^k$  such that  $Q_2^T (\mathcal{A}^T y) Q_2 \succ 0$ , which is guaranteed by Assumption 2. From Lemma 1, the vector  $\hat{y}$  satisfies  $Q_2^T (\mathcal{A}^T \hat{y}) Q_2 = I \succ 0$ .

We will now derive an expression for  $\text{tcone}(\check{y}, Y)$ .

**Theorem 6** Assume that  $\check{y}$  satisfies  $\lambda_{\max}(\mathcal{A}^T \check{y} - C) = 0$  and there exists a Slater point  $y_s$  such that  $\lambda_{\max}(\mathcal{A}^T y_s - C) < 0$ . Then

$$\text{tcone}(\check{y}, Y) = \{d \in \mathbb{R}^k : Q_2^T (\mathcal{A}^T d) Q_2 \preceq 0\} \quad (13)$$

where  $Q_2$  is the nullspace of  $(\mathcal{A}^T \check{y} - C)$ .

**Proof:** We have

$$\begin{aligned} \text{tcone}(\check{y}, Y) &= \{d \in \mathbb{R}^k : d^T v \leq 0, \forall v \in \text{ncone}(\check{y}, Y)\} \\ &= \{d \in \mathbb{R}^k : d^T (\mathcal{A}^T (Q_2 V Q_2^T)) \leq 0, \forall V \succeq 0\} \text{ (from Theorem 5)} \\ &= \{d \in \mathbb{R}^k : (Q_2^T (\mathcal{A}^T d) Q_2) \bullet V \leq 0, \forall V \succeq 0\} \\ &= \{d \in \mathbb{R}^k : Q_2^T (\mathcal{A}^T d) Q_2 \preceq 0\}. \end{aligned}$$

□

We note that an alternative derivation for the tangent cone for the cone of positive semidefinite matrices can be found in Bonnans and Shapiro [6].

## 5.2 When the nullity of the dual slack matrix equals one

If the nullity of  $\check{S}$  is equal to one, then the dimension of the tangent space is  $k - 1$ , as we show in the next theorem. This is as large as possible, of course.

**Theorem 7** If the minimum eigenvalue of  $\bar{S}$  has multiplicity of one with corresponding eigenvector  $\eta$ , then the constraint  $\eta^T (C - \mathcal{A}^T y) \eta \geq 0$  defines a tangent space of  $Y$  of dimension  $k - 1$  at  $\check{y}$ .

**Proof:** The matrix  $\check{S}$  has an eigendecomposition

$$\check{S} = [Q_1 \ \eta] \begin{bmatrix} \Lambda & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} Q_1^T \\ \eta^T \end{bmatrix} \quad (14)$$

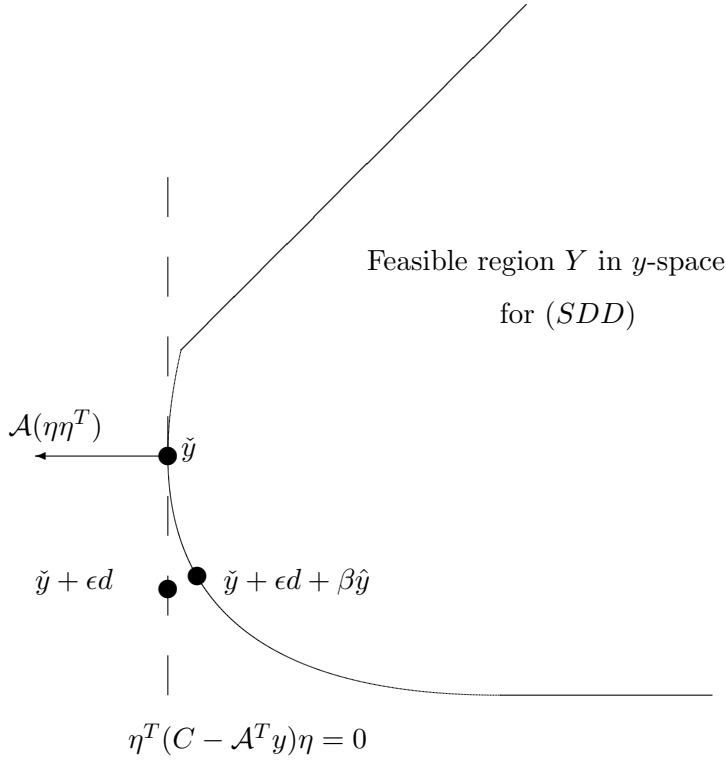


Figure 2: The tangent plane defined by the cutting plane in  $y$ -space

where  $Q = [Q_1 \ \eta]$  is an orthogonal matrix and  $\Lambda$  is a  $(n-1) \times (n-1)$  positive diagonal matrix. For any point  $y$ , define the direction  $d = y - \check{y}$ . The constraint can be written  $\eta^T(\check{S} - A^T d)\eta \geq 0$ , or equivalently as

$$\sum_{i=1}^k d_i \eta^T A_i \eta \leq 0$$

since  $\eta$  is in the nullspace of  $\check{S}$ . The supporting hyperplane  $H$  is defined by the equation

$$\sum_{i=1}^k d_i \eta^T A_i \eta = 0, \quad (15)$$

so  $H = \{\check{y} + d : d \text{ satisfies (15)}\}$ . Moreover from theorem 6,  $d \in \text{tcone}(\check{y}, Y)$  and hence  $d \in \text{tpl}(\check{y}, Y)$ . So,  $\text{tpl}(\check{y}, Y, H)$  has dimension  $k-1$ .  $\square$

Intuitively,  $\lambda_{\max}(A^T y - C)$  is differentiable at  $y = \check{y}$ . So,  $\text{ncone}(\check{y}, Y) = \mathcal{A}(\eta\eta^T)$  (singleton set). Moreover,  $\text{tcone}(\check{y}, Y)$  is the half space  $\sum_{i=1}^k d_i \eta^T A_i \eta \leq 0$  whose boundary is the only supporting hyperplane at  $y = \check{y}$  given by (15). For an illustration of this result, see figure 2.

### 5.3 General values for the nullity

An upper bound on the nullity of an optimal dual slack matrix  $S^*$  is given by Theorem 2. Moreover, we expect the nullity of  $\check{S}$  to be greater than one as Algorithm 1 converges to an optimal solution to (SDD). We will generalize Theorem 7 for the case where the nullity of  $\check{S}$  is greater than one, to get a lower bound on the dimension of the tangent space. Let  $\check{S}$  have an eigendecomposition that is given by (11). We let  $r$  denote the nullity of  $\check{S}$ , and  $\eta$  denotes any vector of norm one in the nullspace of  $\check{S}$ .

The matrix  $Q_2^T(\mathcal{A}^T d)Q_2$  is negative semidefinite if it is zero. This lets us place a lower bound on the dimension of the tangent space.

**Theorem 8** *The dimension of the tangent space to  $Y$  at  $\check{y}$  defined by the valid constraint  $\eta^T(C - \mathcal{A}^T y)\eta \geq 0$  is at least  $k - \binom{r+1}{2}$ .*

**Proof:** The columns of  $Q_2$  give a basis for the nullspace of  $\check{S}$ ; denote these columns as  $p_1, \dots, p_r$ . There are at least  $k - \binom{r+1}{2}$  linearly independent directions  $d$  satisfying the  $\binom{r+1}{2}$  equations

$$p_i^T(\mathcal{A}^T d)p_j = 0, \quad 1 \leq i \leq j \leq r. \quad (16)$$

Note that  $\eta$  is a linear combination of the columns of  $Q_2$ , so any  $d$  satisfying (16) also satisfies  $\eta^T(\mathcal{A}^T d)\eta = 0$ . From Theorem 6, any direction  $d$  satisfying (16) is in the cone of tangents to  $Y$  at  $\check{y}$ . The result follows.  $\square$

Note from Theorem 2 that this result can be quite weak. In Section 6, we give examples where the tangent space has dimension far greater than that implied by theorem 8. It is desirable to choose the vector  $\eta$  to make the dimension of the corresponding tangent plane as large as possible. We now characterize the vectors  $\eta$  which will do this.

Given a  $d = (y - \check{y}) \in \text{tcone}(\check{y}, Y)$ , we have  $v^T d \leq 0$  for every  $v \in \text{ncone}(\check{y}, Y)$ . Therefore, any element  $v \in \text{ncone}(\check{y}, Y)$  gives a valid cutting plane  $v^T y \leq v^T \check{y}$ . The strongest constraints for  $\text{tcone}(\check{y}, Y)$  are those where  $v$  is an extreme ray of  $\text{ncone}(\check{y}, Y)$ . We can find the extreme rays of  $\text{ncone}(\check{y}, Y)$  by finding the extreme points of slices through  $\text{ncone}(\check{y}, Y)$ .

We will illustrate these concepts with the following example:

**Example 2** *Consider*

$$Y = \left\{ y \in \mathbb{R}^2 : S = \begin{bmatrix} y_1 & y_2 & 0 \\ y_2 & y_1 - 3 & 0 \\ 0 & 0 & y_1 - 4 \end{bmatrix} \succeq 0 \right\}.$$

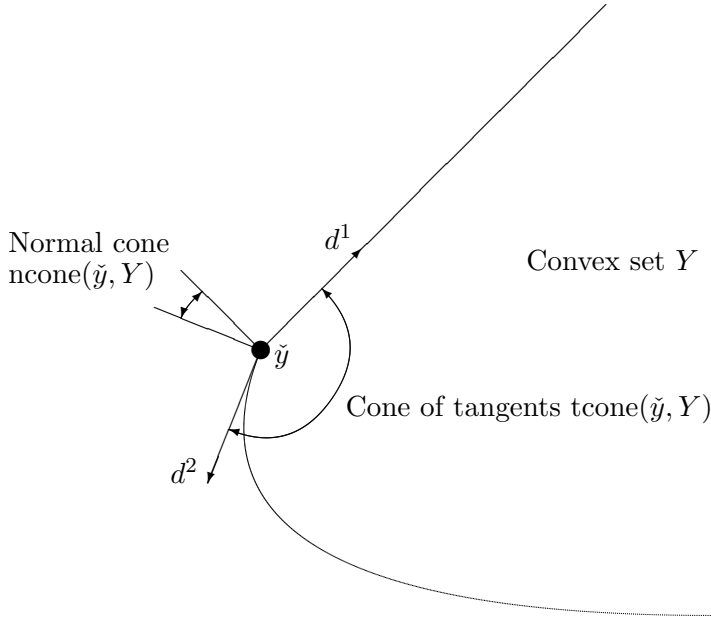


Figure 3: The tangent cone and normal cone for Example 2

Consider the point  $\bar{y} = [3 \ 2]^T$  and let  $\bar{S} = (C - \mathcal{A}^T \bar{y})$ . We have  $\lambda = \lambda_{\min}(\bar{S}) = -1$  with multiplicity 2 and so  $\bar{y} \notin Y$ . One can easily verify that  $\hat{y} = [1 \ 0]^T$ . The point  $\check{y} = (\bar{y} - \lambda \hat{y}) = [4 \ 2]^T \in Y$ . This example is illustrated in figure 3. We have

$$\begin{aligned} tcone(\check{y}, Y) &= \{d \in \mathbb{R}^2 : 5d_1 - 4d_2 \geq 0, \ d_1 \geq 0\} \text{ and} \\ ncone(\check{y}, Y) &= \{v \in \mathbb{R}^2 : 4v_1 + 5v_2 \leq 0, \ v_2 \geq 0\}. \end{aligned}$$

The extreme rays of  $ncone(\check{y}, Y)$  are  $[-5 \ 4]$  and  $[-1 \ 0]$ , respectively. The first ray gives the linear constraint  $5y_1 - 4y_2 \geq 12$  and the tangent space defined by this constraint at  $\check{y}$  is the halfline generated by  $d^2$ . Similarly, the second ray gives the linear constraint  $y_1 \geq 4$  whose tangent space is the halfline generated by  $d^1$ . It is desirable to add constraints corresponding to higher dimensional faces of  $tcone(\check{y}, Y)$ , rather than weaker constraints that are active at  $\check{y}$ , such as  $2y_1 - y_2 \geq 6$ . The latter constraint is obtained from the internal ray  $[-2 \ 1]$  in  $ncone(\check{y}, Y)$ .

Define

$$\Pi = \{v \in ncone(\check{y}, Y) : \hat{y}^T v = 1\} \quad (17)$$

We first show that  $\Pi$  is a closed and bounded set.

**Theorem 9** *The set  $\Pi$  is closed and bounded.*

**Proof:** Using Theorem 5, if  $v \in \text{ncone}(\check{y}, Y)$  then we have  $v = \mathcal{A}(Q_2 V Q_2^T)$  for some  $V \succeq 0$ , so

$$\begin{aligned}\hat{y}^T v &= \hat{y}^T \mathcal{A}(Q_2 V Q_2^T) \\ &= (Q_2^T (\mathcal{A}^T \hat{y}) Q_2) \bullet X \\ &= \text{trace}(X).\end{aligned}$$

Therefore,

$$\Pi = \{\mathcal{A}(Q_2 V Q_2^T) : \text{trace}(V) = 1, V \succeq 0\}.$$

We note that  $\Pi$  is the image under a linear transformation of the closed and bounded set  $\{V \succeq 0 : \text{trace}(V) = 1\}$ , so  $\Pi$  must also be closed and bounded. Alternatively,  $\Pi = \partial \lambda_{\max}(\mathcal{A}^T \check{y} - C)$  (see (5)), that is always a closed and bounded set.  $\square$

It follows from Theorem 9 that we can find all the extreme rays of  $\text{ncone}(\check{y}, Y)$  by solving the semidefinite program  $\min\{g^T v : v \in \Pi\}$  for various values of  $g$ . Further, if the solution for a particular  $g$  is unique then the optimal  $v$  is an extreme ray of  $\text{ncone}(\check{y}, Y)$ . This SDP is easy to solve, requiring just the calculation of the minimum eigenvalue of an  $r \times r$  matrix.

**Theorem 10** *The extreme rays of  $\text{ncone}(\check{y}, Y)$  are vectors of the form  $v = \mathcal{A}(Q_2 u u^T Q_2^T)$ , where  $u$  is an eigenvector of minimum eigenvalue of the matrix  $Q_2^T (\mathcal{A}^T g) Q_2$  for some vector  $g$ .*

**Proof:** Each extreme ray is the solution to a semidefinite program of the form

$$\begin{array}{ll} \min & g^T v \\ \text{subject to} & \mathcal{A}(Q_2 X Q_2^T) - v = 0 \\ & \hat{y}^T v = 1 \\ & X \succeq 0 \end{array} \quad (SP10)$$

for some vector  $g$ . The dual of this problem is

$$\begin{array}{ll} \max & z \\ \text{subject to} & Q_2^T (\mathcal{A}^T y) Q_2 \preceq 0 \\ & \hat{y} z - y = g. \end{array}$$

Substituting  $y = \hat{y} z - g$  into the first constraint and exploiting the facts that  $\mathcal{A}^T \hat{y} = I$  and  $Q_2^T Q_2 = I$ , we obtain the eigenvalue problem

$$\begin{array}{ll} \max & z \\ \text{subject to} & zI \preceq Q_2^T (\mathcal{A}^T g) Q_2. \end{array} \quad (SD10)$$

It follows that the optimal value  $z$  is the smallest eigenvalue of  $Q_2^T (\mathcal{A}^T g) Q_2$ . By complementary slackness, the optimal primal matrix  $X$  must be in the nullspace of the optimal

dual slack matrix, giving the result stated in the theorem.  $\square$

We can use the characterization of the extreme rays given in this theorem to determine explicitly the dimension of the corresponding tangent space of  $Y$ .

**Theorem 11** *Let  $v$  be an extreme ray found by solving (SP10), and assume the nullity of the optimal slack matrix in (SD10) is one. The tangent space defined by the constraint  $v^T y \leq v^T \tilde{y}$  has dimension at least  $k - r$ .*

**Proof:** Let  $u$  be a basis for the null space of the optimal slack matrix for (SD10). From complementary slackness for the (SP10) and (SD10) pair, we have  $v = \mathcal{A}(Q_2 u u^T Q_2^T)$ , rescaling  $u$  if necessary. For a direction  $d$  to be on the tangent space defined by the constraint, we need  $v^T d = 0$  and  $Q_2^T \mathcal{A}^T(d) Q_2 \preceq 0$ . The equality condition can be restated in terms of  $u$  as requiring  $u^T Q_2^T \mathcal{A}^T(d) Q_2 u = 0$ , or equivalently as requiring  $d$  be such that  $u$  is in the nullspace of  $Q_2^T \mathcal{A}^T(d) Q_2$ .

Take  $z$  to be the optimal value of (SP10) and let  $\tilde{d} = \hat{y}z - g$ . From complementary slackness in the pair of semidefinite programs in theorem 10 we have  $v^T \tilde{d} = 0$ . Also  $Q_2^T \mathcal{A}^T(\tilde{d}) Q_2 = (zI - Q_2^T \mathcal{A}^T(g) Q_2) \preceq 0$ . Hence it is clear that  $\tilde{d}$  is in the tangent space defined by the constraint. The vector  $u$  is in the null space of  $Q_2^T \mathcal{A}^T(\tilde{d}) Q_2$ . We now exploit the hypothesis on the nullity of  $Q_2^T \mathcal{A}^T(\tilde{d}) Q_2$ . Without loss of generality, consider the eigendecomposition

$$Q_2^T \mathcal{A}^T(\tilde{d}) Q_2 = [U \ u] \begin{bmatrix} \Lambda & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} U^T \\ u^T \end{bmatrix}$$

where  $\Lambda \prec 0$ . We choose any vector of the form  $d = \tilde{d} + \alpha d'$ , where  $d'$  is chosen so that  $u$  is in the nullspace of  $Q_2^T \mathcal{A}^T(d') Q_2$  and  $\alpha > 0$  is a sufficiently small parameter determined below. It is clear such a choice of  $d'$  will guarantee that  $u$  is in the nullspace of  $Q_2^T \mathcal{A}^T(d) Q_2$ . Now

$$Q_2^T \mathcal{A}^T(d) Q_2 = [U \ u] \begin{bmatrix} \Lambda + \alpha V & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} U^T \\ u^T \end{bmatrix}$$

where  $V = U^T Q_2^T \mathcal{A}^T(d') Q_2 U$ . Since  $\Lambda \prec 0$ , it is clear that the matrix  $Q_2^T \mathcal{A}^T(d) Q_2 \preceq 0$  for an appropriate choice of  $\alpha > 0$ .

The matrix  $Q_2^T \mathcal{A}^T(d') Q_2$  is  $r \times r$ , so  $d' \in \mathbb{R}^k$  must satisfy  $r$  homogeneous equations for  $u$  to be in the nullspace of  $Q_2^T \mathcal{A}^T(d) Q_2$ . It follows that the dimension of the set of possible  $d'$  is at least  $k - r$ , giving the result.  $\square$

This theorem is useful in that it gives us an easy method to distinguish between different vectors  $Q_2 u$  in the nullspace of the dual slack matrix  $\check{S}$ . Algorithmically, we could randomly generate vectors  $g$  and check whether the smallest eigenvalue of  $Q_2^T \mathcal{A}^T(g) Q_2$  has multiplicity equal to one. If so, then we obtain a strong inequality. Note that the matrix  $Q_2^T \mathcal{A}^T(g) Q_2$

is full rank, almost surely, since the columns of  $Q_2$  are linearly independent and since Assumption 2 holds. It follows that the minimum eigenvalue will have multiplicity one almost surely.

Our idea of choosing the extreme points of the subdifferential in constructing the separating hyperplane is complementary to the notion of choosing the minimum norm subgradient from the subdifferential (see Chapter IX in Hiriart-Urruty and Lemaréchal [18]). The negative of the minimum norm subgradient gives the steepest descent direction and it is commonly employed in subgradient and bundle methods for nonsmooth optimization. The problem of finding the minimum norm subgradient can be written as

$$\begin{aligned} \min \quad & v^T v \\ \text{subject to} \quad & v = \mathcal{A}(Q_2 X Q_2^T) \\ & \hat{y}^T v = 1 \\ & X \succeq 0. \end{aligned} \tag{18}$$

The solution  $v$  to (18) usually occurs at an interior point of the feasible region, i.e., at an internal ray of  $\text{ncone}(\check{y}, Y)$ . Consequently, as we show in example 6.2, the generated hyperplane defines a tangent space of low dimension.

## 6 Examples of the tangent space

In this section, we look at examples of the tangent spaces for some semidefinite programs. The positive orthant  $\mathbb{R}_+^n$  can be expressed as the feasible region of a semidefinite program. This simple SDP is the subject of section 6.1. A more complicated example with a large value of  $r$  is the subject of section 6.2.

### 6.1 First example

Take

$$C = 0, \quad A_i = -e_i e_i^T, \quad i = 1, \dots, n.$$

The dual slack matrix is  $S = \text{Diag}(y)$ . This gives  $Y = \mathbb{R}_+^n$ , and  $k = n$ . Thus,  $Y$  is polyhedral and so tangent spaces correspond to faces. Assumption 2 is satisfied and  $\hat{y} = -e$ .

Any positive semidefinite dual slack matrix which has nullity  $r > 0$  gives rise to constraints of any dimension between  $k - 1$  and  $k - r$ . The vector  $\check{y}$  from (10) is a nonnegative vector with  $r$  components equal to zero. The corresponding  $\check{S}$  is a diagonal matrix. The vectors  $\eta$  in the nullspace of  $\check{S}$  are those satisfying  $\eta_i \check{y}_i = 0$  for each component  $i$ . The constraint given in (8) is then  $\sum_{i=1}^n \eta_i^2 y_i \geq 0$ .

In particular, if just  $\check{y}_n = 0$  and all other  $\check{y}_i > 0$  then the only possible  $\eta$  is a multiple of  $e_n$ , which gives the facet-defining constraint  $y_n \geq 0$ .

If several components  $\check{y}_{n-r+1}, \dots, \check{y}_n$  are zero, we can take  $\eta = e_i$ , for some  $n - r + 1 \leq i \leq n$  to give the  $k - 1$  dimensional tangent space defined by the hyperplane  $y_i = 0$ . We

can also take  $\eta = \sum_{i=n-r+1}^n e_i$ , giving the  $k - r$  dimensional tangent space defined by the hyperplane  $\sum_{i=n-r+1}^n y_i = 0$ . Taking  $g$  in (SP10) with  $g_n < g_i$ ,  $i = n - r + 1, \dots, n - 1$  gives  $v = -e_n$ , leading again to the  $k - 1$  dimensional tangent space defined by the hyperplane  $y_n = 0$ .

## 6.2 Second example

Take

$$C = ee^T - I, \quad A_i = -e_i e_i^T, \quad i = 1, \dots, n.$$

This is a non-polyhedral example with  $k = n$  and  $\hat{y} = -e$ . We take  $\check{y} = e$ , so  $\check{S} = ee^T$  and  $r = \text{nullity}(\check{S}) = n - 1$ . Thus,  $k - r = 1$ .

Any  $\eta = e_i - e_j$  with  $i \neq j$  is in the nullspace of  $\check{S}$ . The constraint becomes  $d_i + d_j \geq 0$  for any direction  $d$  from  $\check{y}$ . If  $n = 2$ , then  $r = 1$  and by theorem 7 the hyperplane  $d_1 + d_2 = 0$  defines a tangent space of dimension 1. In this case, we have  $\text{tcone}(\check{y}, Y) = \{d \in \mathbb{R}^2 : d_1 + d_2 \geq 0\}$ .

Suppose,  $n = 3$ . In this case  $r = 2$ . Moreover, using MAPLE we find that

$$\begin{aligned} \text{tcone}(\check{y}, Y) &= \left\{ d \in \mathbb{R}^3 : \begin{bmatrix} d_1 + d_2 + 4d_3 & \sqrt{3}(d_1 - d_2) \\ \sqrt{3}(d_1 - d_2) & 3(d_1 + d_2) \end{bmatrix} \succeq 0 \right\}, \\ \text{ncone}(\check{y}, Y) &= \left\{ v \in \mathbb{R}^3 : \begin{bmatrix} -\frac{3}{2}v_3 & \frac{\sqrt{3}}{2}(-v_1 + v_2) \\ \frac{\sqrt{3}}{2}(-v_1 + v_2) & -v_1 - v_2 + \frac{v_3}{2} \end{bmatrix} \succeq 0 \right\}, \end{aligned}$$

and the feasible region of (SP10) is

$$\Pi = \left\{ v \in \mathbb{R}^3 : v_1 + v_2 + v_3 = -1, \begin{bmatrix} -\frac{3}{2}v_3 & \frac{\sqrt{3}}{2}(-v_1 + v_2) \\ \frac{\sqrt{3}}{2}(-v_1 + v_2) & -v_1 - v_2 + \frac{v_3}{2} \end{bmatrix} \succeq 0 \right\}.$$

Consider the hyperplane  $d_1 + d_2 = 0$ . The tangent space defined by this hyperplane is the half-line  $\{d \in \mathbb{R}^3 : d_1 = 0, d_2 = 0, d_3 \geq 0\}$ , that is of dimension 1. More importantly, there are no facet defining hyperplanes of dimension 2 due to the nonpolyhedral nature of  $\text{tcone}(\check{y}, Y)$ . The choice of  $g = [1 \ 1 \ 0]$  in (SP10) gives the hyperplane  $d_1 + d_2 = 0$ . The minimum norm subgradient obtained from the solution to (18) gives the hyperplane  $d_1 + d_2 + d_3 = 0$ , that intersects  $\text{tcone}(\check{y}, Y)$  at  $[0 \ 0 \ 0]$  and defines a tangent space of dimension 0. Similarly, for  $n \geq 3$  every hyperplane  $d_i + d_j = 0$ ,  $i \neq j$  defines a tangent space of dimension  $n - 2$ , since we can increase the values of  $d_l$ ,  $l \neq i, j$  and stay on the hyperplane as well as  $\text{tcone}(\check{y}, Y)$ . This is strictly larger than  $k - r$  for  $n \geq 4$ .

Now consider  $n = 3$  and the hyperplane  $d_1 + d_2 + 4d_3 = 0$ . The tangent space defined by this hyperplane is the halfline  $\{d \in \mathbb{R}^3 : d_1 \geq 0, d_2 = d_1, d_3 = -2d_1\}$ , that is of dimension 1. For  $n = 4$ , we have

$$\text{tcone}(\check{y}, Y) = \left\{ d \in \mathbb{R}^4 : \begin{bmatrix} 6(d_1 + d_2) & 2\sqrt{3}(d_1 - d_2) & \sqrt{6}(d_1 - d_2) \\ 2\sqrt{3}(d_1 - d_2) & 2(d_1 + d_2 + 4d_3) & \sqrt{2}(d_1 + d_2 - 2d_3) \\ \sqrt{6}(d_1 - d_2) & \sqrt{2}(d_1 + d_2 - 2d_3) & (d_1 + d_2 + d_3 + 9d_4) \end{bmatrix} \succeq 0 \right\}$$

and the hyperplane  $d_1 + d_2 + 4d_3 = 0$  defines the tangent space  $\{d \in \mathbb{R}^4 : d_1 = 0, d_2 = 0, d_3 = 0, d_4 \geq 0\}$ , that is again of dimension 1. For  $n \geq 4$ , the hyperplane  $d_i + d_j + 4d_k \geq 0$ ,  $i \neq j \neq k$  defines a tangent space of dimension  $n - 3$ .

Now assume  $n \geq 4$ . We give a supporting hyperplane that defines a tangent space of dimension 0. Let  $\eta$  be the following vector in the nullspace of  $\check{S}$ :

$$\eta_i = \begin{cases} \lfloor \frac{n}{2} \rfloor & \text{for } i = 1, \dots, \lceil \frac{n}{2} \rceil \\ -\lceil \frac{n}{2} \rceil & \text{for } i = \lceil \frac{n}{2} \rceil + 1, \dots, n \end{cases}$$

Since  $n \geq 4$ , at least two components of  $\eta$  are equal to  $\lfloor \frac{n}{2} \rfloor$  and at least two are equal to  $-\lceil \frac{n}{2} \rceil$ . The constraint for the tangent plane becomes  $\sum_{i=1}^n d_i \eta_i^2 = 0$ . For instance for  $n = 4$ , we have  $\eta = [1 \ 1 \ -1 \ -1]$  which gives the hyperplane  $d_1 + d_2 + d_3 + d_4 = 0$ . Moreover, this is the hyperplane returned by the minimum norm subgradient that is the solution to (18). It can be shown that this hyperplane defines a tangent space of dimension 0 which is strictly smaller than  $k - r$ .

## 7 Adding nonpolyhedral cutting surfaces

We expect the multiplicity of  $\lambda_{\max}(\mathcal{A}^T y - C)$  to grow as Algorithm 1 converges to an optimal solution to (SDD). We will exploit this multiplicity to generate nonpolyhedral cutting surfaces in this section. Let  $\bar{y} \in \mathbb{R}^k$  and let  $P \in \mathbb{R}^{n \times r}$  be an orthonormal matrix containing the eigenspace of  $\lambda_{\max}(\mathcal{A}^T \bar{y} - C)$ . Using the second expression for the subdifferential (5), we derive the following expression for the directional derivative (see Overton [34])

$$D\lambda_{\max}(\mathcal{A}^T \bar{y} - C, d) = \lambda_{\max}(P^T \mathcal{A}^T d P). \quad (19)$$

for  $\lambda_{\max}(\mathcal{A}^T y - C)$  at  $y = \bar{y}$  along the direction  $d \in \mathbb{R}^k$ . Using the convexity of  $\lambda_{\max}(\mathcal{A}^T y - C)$  and setting  $d = y - \bar{y}$  we have

$$\begin{aligned} \lambda_{\max}(\mathcal{A}^T y - C) &\geq \lambda_{\max}(\mathcal{A}^T \bar{y} - C) + \lambda_{\max}(P^T \mathcal{A}^T d P) \\ &= \lambda_{\max}(\mathcal{A}^T \bar{y} - C) + \lambda_{\max}(P^T (\mathcal{A}^T y - C) P) \\ &\quad - \lambda_{\max}(P^T (\mathcal{A}^T \bar{y} - C) P) \\ &= \lambda_{\max}(P^T (\mathcal{A}^T y - C) P). \end{aligned} \quad (20)$$

Therefore, one can add the convex constraint  $\lambda_{\max}(P^T (\mathcal{A}^T y - C) P) \leq 0$ , which can also be expressed as the following semidefinite cutting surface

$$P^T (C - \mathcal{A}^T y) P \succeq 0 \quad (21)$$

that is satisfied by all feasible  $y \in \mathbb{R}^k$ . Defining  $\check{y}$  as in (10), we have, analogous to Theorem 4, that  $P^T (C - \mathcal{A}^T \check{y}) P = 0$ , since the columns of  $P$  give a basis for the nullspace of  $(C - \mathcal{A}^T \check{y})$ . It follows from Theorem 6 that the set of feasible directions  $d$  at  $\check{y}$  which satisfy constraint (21) is exactly the cone of tangents.

One can add this nonpolyhedral cutting surface to the dual relaxation. This gives a conic relaxation of (SDD) over linear and smaller dimensional semidefinite cones that is solved in every iteration — see Krishnan et al. [41] for more details. Of course, the practical disadvantage of adding such a nonpolyhedral cutting surface is that  $r$  can get very large. In this case, the complexity of solving the conic relaxation is comparable that of solving (SDD) with a primal-dual interior point method. One way to reduce the complexity is to add a nonpolyhedral constraint that uses only a subset of the columns of  $P$ , with the concomitant disadvantage that the cone of tangents is not captured as accurately.

Another technique for generating nonpolyhedral cutting surfaces is the following: The  $\epsilon$  subdifferential (see Chapter XI in Hiriart-Urruty and Lemaréchal [18]) for  $\bar{\lambda} = \lambda_{\max}(\mathcal{A}^T y - C)$  at  $y = \bar{y}$  is given by

$$\partial_{\epsilon} \lambda_{\max}(\mathcal{A}^T \bar{y} - C) = \{ \mathcal{A}(PV P^T) : \text{trace}(V) = 1, V \succeq 0 \} \quad (22)$$

where  $V \in \mathcal{S}^r$ ; and  $P \in \mathbb{R}^{n \times r}$  is an orthogonal matrix, whose columns  $p_i$ ,  $i = 1, \dots, r$  are eigenvectors corresponding to the eigenvalues that are within an  $\epsilon$  of  $\bar{\lambda}$ , i.e.,  $(\mathcal{A}^T \bar{y} - C)p_i = (\bar{\lambda} - \epsilon_i)p_i$ ,  $i = 1, \dots, r$ , with  $\epsilon_i \leq \epsilon$ .

Using the  $\epsilon$  subdifferential, we can derive the following expression

$$D\lambda_{\max, \epsilon}(\mathcal{A}^T \bar{y} - C, d) = \lambda_{\max}(P^T \mathcal{A}^T d P) \quad (23)$$

for the  $\epsilon$  directional derivative (see XI.2 in Hiriart-Urruty and Lemaréchal [18]) in a manner similar to (19). Using the definition of the  $\epsilon$  directional derivative, we have with  $y = \bar{y} + d$ :

$$\begin{aligned} \lambda_{\max}(\mathcal{A}^T y - C) &\geq \lambda_{\max}(\mathcal{A}^T \bar{y} - C) + D\lambda_{\max, \epsilon}(\mathcal{A}^T \bar{y} - C, d) - \epsilon \\ &= \lambda_{\max}(\mathcal{A}^T \bar{y} - C) + \lambda_{\max}(P^T (\mathcal{A}^T y - C) P - \Lambda) - \epsilon \\ &\geq \lambda_{\max}(\mathcal{A}^T \bar{y} - C) + \lambda_{\max}(P^T (\mathcal{A}^T y - C) P) - \lambda_{\max}(\Lambda) - \epsilon \\ &= \lambda_{\max}(P^T (\mathcal{A}^T y - C) P) - \epsilon \end{aligned} \quad (24)$$

where  $\epsilon > 0$  is an appropriate scalar and  $\Lambda \in \mathcal{S}^r$  is a diagonal matrix containing the eigenvalues whose eigenvectors are in  $P$ . The 2nd inequality in (24) follows from Weyl's monotonicity theorem for symmetric matrices (see Horn and Johnson [19]). Consider

$$P^T (C - \mathcal{A}^T y) P \succeq 0 \quad (25)$$

that gives a valid semidefinite cutting surface. To illustrate the strength of this constraint, consider points  $y$  which are in the region where the directional derivative gives a good approximation to the function. It follows from (24) that any such points which are infeasible will violate (25) (assuming  $\epsilon > 0$  is small and Weyl's monotonicity theorem is providing a tight inequality). These nonpolyhedral cutting surfaces are employed in the interior point cutting plane algorithms of Krishnan et al. [41] and Oskoorouchi and Goffin [32]. As discussed previously, it is important to keep  $r$  small to keep the complexity of solving the conic relaxation of the dual problem small. An upper bound on  $r$  given by Theorem 2 is maintained in [41] and [32]. Moreover, when  $\epsilon$  is a large positive number, the semidefinite cutting surface (25) becomes the entire boundary of the dual feasible region  $Y$ .

## 8 Conclusions

The results of section 4 show that interior point cutting plane approaches to the solution of semidefinite programming problems have attractive theoretical complexity. Such algorithms are more attractive for larger scale problems where standard interior point methods become impractical. Computational results with the interior point cutting plane algorithm described in this paper can be found in Krishnan et al. [41].

Our primary focus in this paper is the design of good separation oracles that return cutting planes that support the feasible region of the dual semidefinite program. Furthermore, we introduce a concept called the *tangent space induced by a supporting hyperplane* that measures the strength of a cutting plane. The results of section 5 show that a facet defining cutting plane can always be found if the nullity of the dual slack matrix is one. Further, for higher values of the nullity, cutting planes that induce a higher dimensional tangent space can be found by determining the smallest eigenvalue and corresponding eigenvector of a matrix. When the nullity is larger, it is of interest to determine a set of cutting planes that work well together to give a good approximation to the cone of tangents.

The cone of tangents can be captured exactly by a nonpolyhedral cutting surface, as we show in section 5. We also describe other oracles returning nonpolyhedral cutting surfaces, which have been employed in the conic interior point cutting algorithms of Krishnan et al. [41] and Oskoorouchi and Goffin [32]. Adding many cuts of this type over a number of iterations will lead to a large conic program, so some pruning is needed. It is also possible to combine new cutting planes/surfaces with old ones in a nonpolyhedral manner; this is described in Krishnan et al. [22, 41] and is also one of the motivations behind the spectral bundle method of Helmberg and Rendl [15].

## References

- [1] F. Alizadeh, J.-P. Haeberly, and M. L. Overton. Complementarity and nondegeneracy in semidefinite programming. *Mathematical Programming*, 77(2):111–128, 1997.
- [2] K. M. Anstreicher. On Vaidya’s volumetric cutting plane method for convex programming. *Mathematics of Operations Research*, 22(1):63–89, 1997.
- [3] K. M. Anstreicher. Towards a practical volumetric cutting plane method for convex programming. *SIAM Journal on Optimization*, 9(1):190–206, 1999.
- [4] K. M. Anstreicher. The volumetric barrier for semidefinite programming. *Mathematics of Operations Research*, 25(3):365–380, 2000.

- [5] A. Ben-Tal and A. Nemirovski. *Lectures on Modern Convex Optimization: Analysis, Algorithms, and Engineering Applications*, volume 2 of *MPS/SIAM Series on Optimization*. SIAM, Philadelphia, 2001.
- [6] J. F. Bonnans and A. Shapiro. *Perturbation Analysis of Optimization Problems*. Springer-Verlag, New York, 2001.
- [7] B. Borchers and J. G. Young. Implementation of a primal-dual method for SDP on a shared memory parallel architecture. *Computational Optimization and Applications*, 37(3):355–369, 2007.
- [8] S. Burer and R. D. C. Monteiro. A nonlinear programming algorithm for solving semidefinite programs via low-rank factorization. *Mathematical Programming*, 95(2):329–357, 2003.
- [9] A. R. Conn, N. I. M. Gould, and P. L. Toint. *Trust Region Methods*. SIAM, Philadelphia, 2000.
- [10] E. de Klerk. *Aspects of Semidefinite Programming: Interior Point Algorithms and Selected Applications*, volume 65 of *Applied Optimization*. Kluwer Academic Publishers, Dordrecht, Netherlands, 2002.
- [11] M. Grötschel, L. Lovasz, and A. Schrijver. *Geometric Algorithms and Combinatorial Optimization*. Springer-Verlag, Berlin, Germany, 1988.
- [12] C. Helmberg. Semidefinite programming for combinatorial optimization. Technical Report ZR-00-34, TU Berlin, Konrad-Zuse-Zentrum, Berlin, October 2000. Habilitationsschrift.
- [13] C. Helmberg. Numerical evaluation of SBmethod. *Mathematical Programming*, 95(2):381–406, 2003.
- [14] C. Helmberg and K. Kiwiel. A spectral bundle method with bounds. *Mathematical Programming*, 93(2):173–194, 2002.
- [15] C. Helmberg and F. Rendl. A spectral bundle method for semidefinite programming. *SIAM Journal on Optimization*, 10(3):673–696, 2000.
- [16] R. Hettich and K. O. Kortanek. Semi-infinite programming: theory, methods and applications. *SIAM Review*, 35(3):380–429, 1993.
- [17] J.-B. Hiriart-Urruty and C. Lemaréchal. *Convex Analysis and Minimization Algorithms I*. Springer-Verlag, Berlin, 1993.
- [18] J.-B. Hiriart-Urruty and C. Lemaréchal. *Convex Analysis and Minimization Algorithms II*. Springer-Verlag, Berlin, 1993.

- [19] R. A. Horn and C. Johnson. *Matrix Analysis*. Cambridge University Press, Cambridge, 1985.
- [20] K. Krishnan. *Linear programming approaches to semidefinite programming problems*. PhD thesis, Mathematical Sciences, Rensselaer Polytechnic Institute, Troy, NY 12180, July 2002. <http://www4.ncsu.edu/~kksivara/rpithes.pdf>.
- [21] K. Krishnan and J. E. Mitchell. Semi-infinite linear programming approaches to semidefinite programming (SDP) problems. In P. M. Pardalos and H. Wolkowicz, editors, *Novel approaches to hard discrete optimization problems*, volume 37 of *Fields Institute Communications Series*, pages 123–142. AMS, 2003.
- [22] K. Krishnan and J. E. Mitchell. A unifying framework for several cutting plane methods for semidefinite programming. *Optimization Methods and Software*, 21(1):57–74, 2006.
- [23] K. Krishnan and J.E. Mitchell. A semidefinite programming based polyhedral cut-and-price approach for the maxcut problem. *Computational Optimization and Applications*, 33(1):51–71, 2006.
- [24] M. Laurent and F. Rendl. Semidefinite programming and integer programming. In K. Aardal, G. Nemhauser, and R. Weismantel, editors, *Handbook on Discrete Optimization*, volume 12 of *Handbooks in Operations Research and Management Sciences*, pages 393–514. Elsevier, Amsterdam, The Netherlands, 2005.
- [25] J. E. Mitchell. Polynomial interior point cutting plane methods. *Optimization Methods and Software*, 18(5):507–534, 2003.
- [26] J. E. Mitchell and M. J. Todd. Solving combinatorial optimization problems using Karmarkar’s algorithm. *Mathematical Programming*, 56(3):245–284, 1992.
- [27] R. D. C. Monteiro. First- and second-order methods for semidefinite programming. *Mathematical Programming*, 97(1–2):209–244, 2003.
- [28] M.V. Nayakkankuppam. Solving large-scale semidefinite programs in parallel. *Mathematical Programming*, 109(2-3):477–504, 2007.
- [29] G. L. Nemhauser and L. A. Wolsey. *Integer and Combinatorial Optimization*. John Wiley, New York, 1988.
- [30] A. S. Nemirovsky and D. B. Yudin. *Problem Complexity and Method Efficiency in Optimization*. John Wiley, 1983.
- [31] Y. E. Nesterov and A. S. Nemirovskii. *Interior Point Polynomial Methods in Convex Programming: Theory and Algorithms*. SIAM Publications. SIAM, Philadelphia, USA, 1993.

- [32] M. R. Oskoorouchi and J.-L. Goffin. A matrix generation approach for eigenvalue optimization. *Mathematical Programming*, 109(1):155–179, 2007.
- [33] M. R. Oskoorouchi and J. E. Mitchell. A second-order cone cutting surface method: complexity and application. Technical report, College of Business Administration, California State University San Marcos, California, May 2005. <http://www.rpi.edu/~mitchj/papers/socr.html>.
- [34] M. Overton. Large-scale optimization of eigenvalues. *SIAM Journal on Optimization*, 2(1):88–120, 1992.
- [35] 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.
- [36] G. Pataki. The geometry of semidefinite programming. In H. Wolkowicz, R. Saigal, and L. Vandenberghe, editors, *Handbook of Semidefinite Programming*, pages 29–65. Kluwer Academic Publishers, Dordrecht, The Netherlands, 2000.
- [37] G. Pataki. On the closedness of the linear image of a closed convex cone. *Mathematics of Operations Research*, 32(2):395–412, 2007.
- [38] S. Ramaswamy and J. E. Mitchell. A long step cutting plane algorithm that uses the volumetric barrier. Technical report, DSES, Rensselaer Polytechnic Institute, Troy, NY 12180, June 1995.
- [39] H. D. Sherali and B. M. P. Fraticelli. Enhancing RLT relaxations via a new class of semidefinite cuts. *Journal of Global Optimization*, 22(1–4):233–261, 2002.
- [40] K. Krishnan Sivaramakrishnan. A parallel interior point decomposition algorithm for block angular semidefinite programs. Technical report, Department of Mathematics, North Carolina State University, Raleigh, NC 27695-8205, August 2007. <http://www4.ncsu.edu/~kksivara/publications/parallel-conic-blockangular.pdf>.
- [41] K. Krishnan Sivaramakrishnan, G. Plaza, and T. Terlaky. A conic interior point decomposition approach for large scale semidefinite programming. Technical report, Department of Mathematics, North Carolina State University, Raleigh, NC 27695-8205, December 2005. <http://www4.ncsu.edu/~kksivara/publications/conic-ipm-decomposition.pdf>.
- [42] M. J. Todd and Y. Ye. A centered projective algorithm for linear programming. *Mathematics of Operations Research*, 15(3):508–529, 1990.
- [43] P. M. Vaidya. A new algorithm for minimizing convex functions over convex sets. *Mathematical Programming*, 73(3):291–341, 1996.

- [44] M. Yamashita, K. Fujisawa, and M. Kojima. SDPARA: Semidefinite programming algorithm parallel version. *Parallel Computing*, 29(8):1053–1067, 2003.
- [45] Y. Zhang, R. A. Tapia, and J. E. Dennis. On the superlinear and quadratic convergence of primal–dual interior point linear programming algorithms. *SIAM Journal on Optimization*, 2(2):304–324, 1992.