Properties of a Cutting Plane Method for Semidefinite Programming[1]

Kartik Krishnan
Department of Computational and Applied Mathematics
Rice University
Houston, TX, 77005-1892
kartik@caam.rice.edu
http://www.caam.rice.edu/~kartik

John E. Mitchell
Mathematical Sciences
Rensselaer Polytechnic Institute
Troy, NY 12180
mitchj@rpi.edu
http://www.rpi.edu/~mitchj

May 14, 2003

Abstract

A semidefinite programming problem is a nonsmooth optimization problem, so it can be solved using a cutting plane approach. In this paper, we analyze properties of such an algorithm. We discuss characteristics of good polyhedral representations for the semidefinite program. We show that the complexity of an interior point cutting plane approach based on a semi-infinite formulation of the semidefinite program has complexity comparable with that of a direct interior point solver. We show that cutting planes can always be found efficiently that support the feasible region. Further, we characterize the supporting hyperplanes that give high dimensional tangent planes, and show how such supporting hyperplanes can be found efficiently.

Keywords: Semidefinite programming, column generation, cutting plane methods, tangent space.

[1]Research supported in part by NSF grant number CCR–9901822

1 Introduction

Semidefinite programming problems arise in many different settings. The recent interest in semidefinite programming was spurred by the discovery of very good relaxations for combinatorial optimization problems as well as the development of efficient interior point methods for their solution. Semidefinite programming also has important applications in control theory. For good surveys on various aspects of semidefinite programming, see Laurent and Rendl [19], Helmberg [10], Todd [30], Vandenberghe and Boyd [33], the Handbook of Semidefinite Programming [34], and the webpage maintained by Helmberg [8].

A semidefinite programming problem is a convex optimization problem. It can be expressed in terms of variables that are arranged in matrices, with a linear objective function, linear constraints, and requirements that the matrices be positive semidefinite. There are direct polynomial time interior point methods to solve these problems, but there are practical limitations on the size of problems that these methods can currently handle. Therefore, there has been recent interest in developing alternative approaches to the solution of semidefinite programs which can handle large scale problems more effectively. These alternatives include the bundle method of Helmberg and Rendl [11, 9, 10], the nonlinear programming approach of Burer, Monteiro and Zhang [5], and the cutting plane methods of Oskoorouchi and Goffin [25] and Krishnan and Mitchell [15, 16, 17].

In this paper, we describe some of the properties of the algorithm presented in [15, 16]. This method is based upon a semi-infinite formulation of a semidefinite program, and it uses a cutting plane method to obtain a polyhedral approximation to the feasible region. We introduce the semi-infinite formulation in §2, we give an optimal polyhedral description in §3, and describe the algorithm in §4.

Our algorithm uses an interior point method to approximately solve the linear programming relaxations that arise at each iteration. The implementation uses a primal-dual predictor-corrector method — see [15, 16, 18] for computational results. Theoretically, it is possible to use a volumetric interior point method to solve the relaxations. As we show in §5, this would result in an algorithm whose theoretical complexity for solving a semidefinite program is slightly smaller than that of a direct interior point method, at least for a problem where the number of constraints is approximately equal to the dimension of the matrix.

The tangent space formed by the constraints generated by the algorithm is the topic of §6. We show that we can always find a linear constraint that supports the feasible region. We show that in some situations, the tangent plane defined by the constraint will have dimension just one less than the dimension of the space, and we give a lower bound on the dimension of the tangent plane defined by the constraint. Further, if the nullity of the active dual slack matrix is r and the primal SDP has k constraints, then a linear constraint can be found that gives a tangent plane of dimension $k - r$. Examples are contained in §7. The strength of nonpolyhedral cutting planes is the subject of §8.

It should be noted that cutting plane methods are usually less efficient in practice than direct interior point methods for the SDP for problems to which the latter methods apply, and typically they are unable to solve an SDP as accurately in comparable time. It is hard enough to get even a few digits of accuracy.

Many SDP problems arise as relaxations of various combinatorial optimization problems. Without loss of generality, let us assume that our problem is a minimization problem. The SDP relaxation serves as a lower bound which can usually be tightened, and so it is not necessary to solve it down to optimality. However we would still like to get a good solution since

- A good lower bound can result in an early termination of the cutting plane approach for the underlying combinatorial optimization problem.

- We also round the SDP solution to generate integer solutions; incidentally they serve as upper bounds in the cutting plane approach. Again, a good solution to the SDP is more likely to generate a good integer solution.

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 ith diagonal entry equal to v_i for $i = 1, \ldots, n$ is denoted $\mathrm{Diag}(v)$. 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 $\mathrm{tr}(AB)$.

We conclude this section with some supplementary notation that will be useful in §6. Consider

$$X = \begin{bmatrix} P_1 & P_2 \end{bmatrix} \begin{bmatrix} V & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} P_1^T \\ P_2^T \end{bmatrix}$$

Let $P = \begin{bmatrix} P_1 & P_2 \end{bmatrix}$. We consider the representations of X, S in the transformed space P as follows :

$$\begin{aligned} \mathring{X} &= P^T X P &= \begin{bmatrix} V & 0 \\ 0 & 0 \end{bmatrix} \\ \mathring{S} &= P^T S P &= \begin{bmatrix} \mathring{S}_{11} & \mathring{S}_{12} \\ \mathring{S}_{12}^T & \mathring{S}_{22} \end{bmatrix} \end{aligned}$$

The transformed space P will be clear from the context.

2 Semi-infinite formulations for semidefinite programming

Consider the semidefinite programming problem

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

with dual

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

where $X, S \in \mathcal{S}^n_+$, C is an $n \times n$ matrix, b and y are k-vectors, and $\mathcal{A}$ is a linear function mapping $\mathcal{S}^n$ to $I\!R^k$. 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 $A_i \bullet X = b_i$, $i = 1, \ldots, k$. 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 *Both (SDP) and (SDD) have strictly feasible solutions, that is, there exists a positive definite matrix X satisfying $\mathcal{A}(X) = b$ and there exists a vector y such that $C - \mathcal{A}^T y$ is positive definite.*

This assumption ensures that we have *strong duality* at optimality, i.e. the objective values of (SDP), and (SDD) are equal, and both problems attain their optimal solution.

Assumption 2 *The matrices A_i, $i = 1, \ldots, k$, are linearly independent in $\mathcal{S}^n$.*

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

$$d^T X d \;=\; dd^T \bullet X \;\geq\; 0 \quad \forall d \in B \tag{1}$$

where B is a compact set, typically $\{d : ||d||_2 \leq 1\}$ or $\{d : ||d||_\infty \leq 1\}$. These are commonly used in *trust region methods* (Conn et al [6]). These constraints are linear inequalities in the matrix variable X, but there is an *infinite* number of them. Thus SDP is a *semi-infinite* linear programming problem in $\mathbb{R}^{\frac{n(n+1)}{2}}$. The term semi-infinite programming derives from the fact that the LP has finitely many variables, with an infinite number of constraints. The survey paper by Hettich and Kortanek [12] discusses theory, algorithms, and applications of semi-infinite programming. One can also deduce that the primal SDP cone $\{X : X \succeq 0\}$ is *convex* from this semi-infinite formulation.

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

Lemma 1 *The symmetric $n \times n$ matrix S is positive semidefinite if and only if $S \bullet M \geq 0$ for all symmetric rank one matrices M.*

We now consider two semi-infinite linear programs $(PSIP)$ and $(DSIP)$ for (SDP) and (SDD) respectively. These formulations follow directly from Corollary 1.

$$\begin{array}{lrcll} \min & C \bullet X & & & \\ \text{subject to} & \mathcal{A}(X) & = & b & \qquad (PSIP) \\ & d^T X d & \geq & 0 \quad \forall d \in B & \end{array}$$

$$\begin{array}{lrcll} \max & b^T y & & & \\ \text{subject to} & \mathcal{A}^T y + S & = & C & \qquad (DSIP) \\ & d^T S d & \geq & 0 \quad \forall d \in B & \end{array}$$

Note that X is $n \times n$ and symmetric, so $(PSIP)$ is a semi-infinite linear program in $\binom{n+1}{2} = \frac{n(n+1)}{2} = O(n^2)$ variables. In contrast, there are k variables in the semi-infinite formulation $(DSIP)$. We have $k \leq \binom{n+1}{2}$ (since the matrices A_i, $i = 1, \ldots, k$ are linearly independent). Therefore, it is more efficient to deal with the dual semi-infinite formulation, since we are dealing with smaller LP's (but see also the discussion at the end of this section). We shall henceforth refer to $(DSIP)$ as (LDD).

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

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

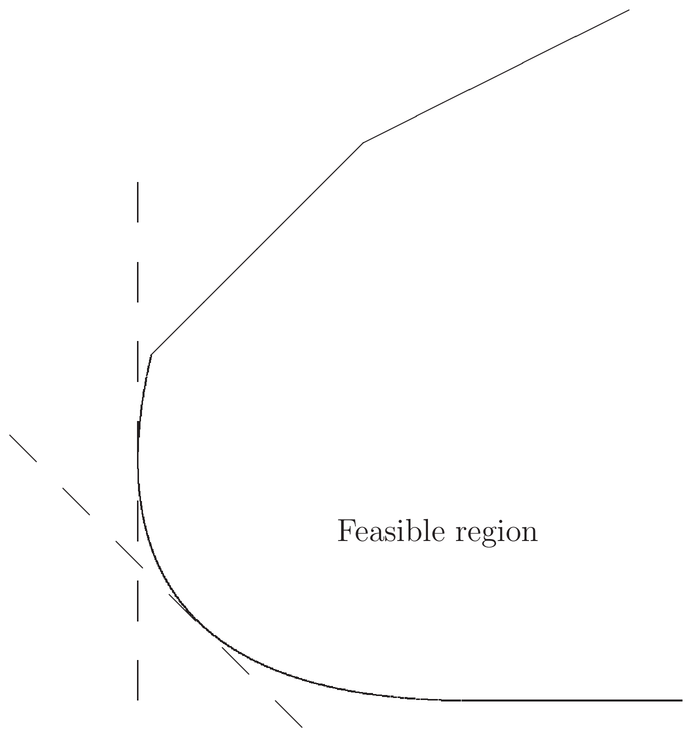

Figure 1: An LP approximation to a semidefinite program. The dashed lines are a polyhedral approximation to the curved boundary of the feasible region.

as illustrated in figure 1. The linear programming dual to (LDR) is a constrained version of (SDP), and it can be expressed in such a way as to make this explicit, as we now show. We have

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

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

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

It follows that the dual problem is

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

This can be rewritten as

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

Theorem 1 *Any feasible solution x to (LPR) will give a feasible solution X to (SDP).*

Proof: This result follows directly from the fact that (LPR) is a constrained version of (SDP). However we present a formal proof. Set $X = \sum_{i=1}^m x_i d_i d_i^T$. From (LPR) it is clear that this X satisfies $\mathcal{A}X = b$. Moreover X is psd. To see this

$$d^T X d \;=\; d^T(\sum_{i=1}^m x_i d_i d_i^T)d \;=\; \sum_{i=1}^m x_i (d_i^T d)^2 \;\geq\; 0 \qquad \forall d$$

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

If X is feasible in (SDP) and (y, S) is feasible in (SDD), they are optimal if and only if they satisfy the complementary slackness relationship $XS = 0$. The duality gap is equal to the trace of this matrix. We show that at the optimal solution to (LDR) and (LPR), we satisfy the reduced complementary slackness condition $\mathrm{tr}(XS) = 0$.

Theorem 2 *Let $\hat{y}$ be optimal for (LDD) and $\hat{x}$ be optimal for (LPR). Let $\hat{X} = \sum_{i=1}^m \hat{x}_i d_i d_i^T$ and $\hat{S} = C - \mathcal{A}^T(\hat{y})$. The matrices $\hat{X}$ and $\hat{S}$ satisfy tr$(\hat{X}\hat{S}) = 0$. Furthermore, if $\hat{S}$ is positive semidefinite then $\hat{X}\hat{S} = 0$ and $\hat{X}$ is optimal for (SDP).*

Proof: We have

$$\begin{aligned} \mathrm{tr}(\hat{X}\hat{S}) &= \mathrm{tr}(\sum_{i=1}^m \hat{x}_i d_i d_i^T (C - \mathcal{A}^T(\hat{y}))) \\ &= \sum_{i=1}^m \hat{x}_i d_i^T (C - \mathcal{A}^T(\hat{y}))d_i \\ &= 0 \end{aligned}$$

from the complementary slackness relationship at optimality for (LPR) and (LDR).

If $\hat{S}$ is positive semidefinite then it is feasible in (SDD). If in addition $\mathrm{tr}(\hat{X}\hat{S}) = 0$, we have $\hat{x}_i d_i^T (C - \mathcal{A}^T(\hat{y}) d_i = 0$ for $i = 1, \ldots, m$. Thus, each d_i for which $\hat{x}_i > 0$ is in the nullspace of $\hat{S}$, so $\hat{X}\hat{S} = 0$ and $\hat{X}$ solves (SDP). $\square$

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 $\mathcal{B} : \mathcal{S}^n \to I\!R^q$ be the null space operator corresponding to $\mathcal{A}$, so the kernel of $\mathcal{B}^T$ is exactly the range of $\mathcal{A}$. From assumption 2, 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 $u \in I\!R^q$. The problem (SDP) can then be written equivalently as

$$\begin{array}{llcccl} \min_{u,X} & C \bullet X_0 - C \bullet \mathcal{B}^T(u) & & & & \\ \text{subject to} & \mathcal{B}^T(u) \;\; + & X & = & X^0 & \qquad (SDPN) \\ & & X & \succeq & 0, & \end{array}$$

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 §5. (A similar nullspace representation of linear programming problems has been analyzed in the interior point literature; see, for example, Todd and Ye [31] and Zhang et al. [35].)

3 A perfect set of constraints

The semi-infinite formulation $(DSIP)$ has a finite relaxation (LDR) for a set of vectors $\{d_i,\, i = 1, \ldots, m\}$. We discuss the perfect set of linear constraints that are required for the SDP.

The optimality conditions for the SDP include primal feasibility, dual feasibility and complementarity $XS = 0$. The complementarity condition implies that X and S commute, and so they share a common share of eigenvectors (see *simultaneous diagonalization* in Horn and Johnson [14]).

Thus, we have (Alizadeh et al [1]):

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

$$X = [Q \;\; R] \begin{bmatrix} \Lambda & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} Q^T \\ R^T \end{bmatrix} \qquad (2)$$

$$S = [Q \;\; R] \begin{bmatrix} 0 & 0 \\ 0 & \Omega \end{bmatrix} \begin{bmatrix} Q^T \\ R^T \end{bmatrix} \qquad (3)$$

hold.

The diagonal matrices Λ, Ω contain the nonzero eigenvalues of X and S in the spectral decompositions (2) and (3) respectively. Also $P = [Q \quad R]$ is an orthogonal matrix that contains the common set of eigenvectors.

We get an upper bound on r, using the following theorem, due to Pataki [26] (also see [1]), on the rank of extreme matrices X in (SDP).

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

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

The spectral decomposition of the optimal solution to (SDP) suggests a set of linear constraints for (LDD), namely the columns of the matrix Q. With this choice, the optimal solution to the linear programming relaxation gives the optimal solution to the original semidefinite program.

Theorem 5 *Let $X^* = Q\Lambda Q^T$ be optimal for (SDP), and let q_i, $i = 1, \ldots, r$ be the columns of Q. Then the optimal solutions to (SDP) and (SDD) are given by the optimal solutions to (4), and its dual: (5).*

$$\begin{array}{lrcl} \max & b^T y & & \\ s.t. & \sum_{j=1}^{k}(q_i^T A_j q_i) y_j & \leq & q_i^T C q_i, \quad i = 1, \ldots, r \end{array} \tag{4}$$

with dual

$$\begin{array}{lrcl} \min & \sum_{i=1}^{r}(q_i^T C q_i) x_i & & \\ s.t. & \sum_{i=1}^{r}(q_i^T A_j q_i) x_i & = & b_j \quad j = 1, \ldots, k \\ & x & \geq & 0 \end{array} \tag{5}$$

Proof: Since (4) is a discretization to (SDD), its dual (5) is a constrained version of (SDP). Thus its optimal value gives an upper bound on the optimal value of (SDP). Thus an optimal solution to (SDP) is also optimal in (5), provided it is feasible in it. The optimal solution X^* to (SDP) is clearly feasible in (5), with $x = \lambda$. This corresponds to (LDR) with $d_i = q_i$, $i = 1, \ldots, r$. $\square$

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

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

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

where the columns of the $n \times m$ matrix D are the vectors $d_i, i = 1, \ldots, m$. If $D = Q$ then we obtain (5).

Problem (LPR) is a constrained version of (SDP). The requirement that M be diagonal in (6) can be relaxed to requiring that M be positive semidefinite, or to requiring that M be positive semidefinite with a specified block-diagonal structure, and a constrained version of (SDP) is still obtained. These alternatives are discussed in Krishnan and Mitchell [17]. The block diagonal relaxation is used by Oskoorouchi and Goffin [25]. The relaxation that M be positive semidefinite, with m small compared to n, is the basis of the spectral bundle method of Helmberg and Rendl [11, 9, 10]. We return to these alternative nonpolyhedral cutting planes in §8.

These alternatives are semidefinite programs, so they are harder to solve than (LPR). One advantage of the relaxation that M be psd is that it will give the optimal solution to (SDP) if the columns of D form a basis for the nullspace of S; the relaxation (LPR) requires that the columns of D give bases for each eigenspace of the optimal X with a positive eigenvalue.

In the LP approach, the number of columns in D can get arbitrarily large. However when this number is more than n, we could instead perform a spectral factorization of $X = DMD^T = P\Lambda P^T$, and replace the columns of D with the set of eigenvectors in P. If the eigenspaces of the current matrix X are close to those of the optimal solution, this refactorization may have the benefit of giving constraints that are close to the set of constraints given in theorem 5.

4 A cutting plane method for semidefinite programming

The goal of cutting plane methods is to find the optimal point in a convex set Y, or to determine that Y is empty. Given a point $\bar{y}$, a separation oracle either tells us that $\bar{y} \in Y$ (in which case we try to improve the objective function) or it returns a separating hyperplane that separates $\bar{y}$ from the set Y. The cuts returned by the oracle at a point $\bar{y} \notin Y$ have the following form :

$$a^T y \quad \geq \quad a^T \bar{y} + \gamma \tag{7}$$

If $\gamma > 0$, the cut is *deep*; if $\gamma < 0$, the cut is *shallow*, and finally if $\gamma = 0$, the cut passes through $\bar{y}$, and we shall refer to this as a *central* cut. Our intuition suggests that deep cuts are better, since they exclude a larger set of points from consideration. The cuts

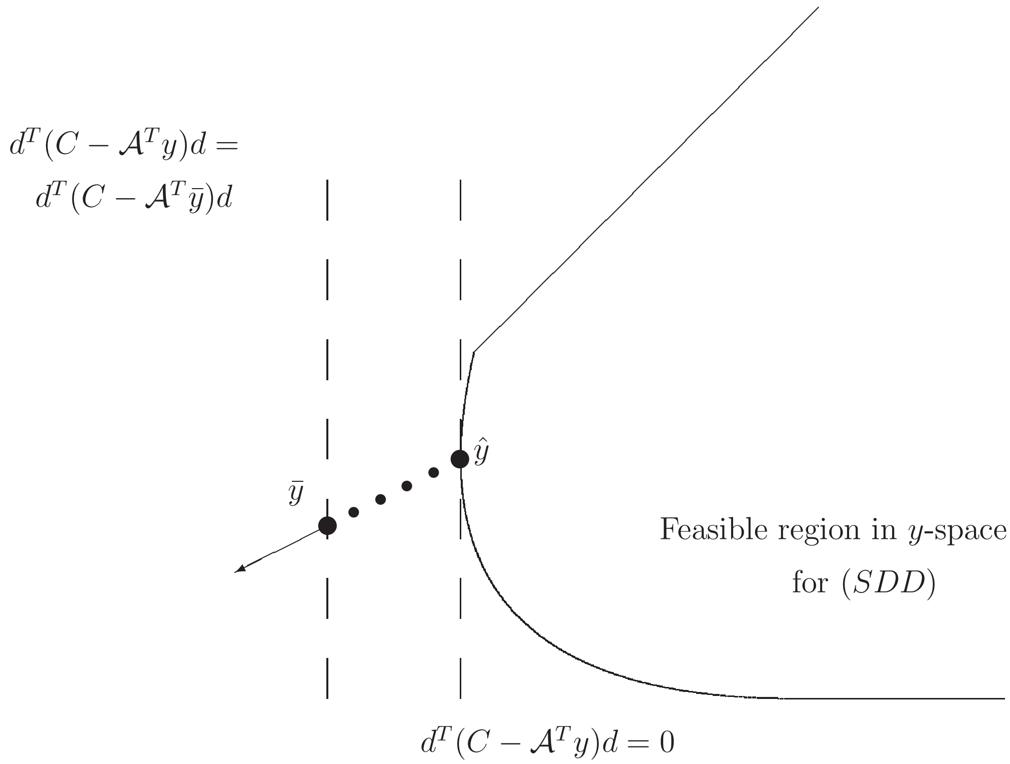

Figure 2: Adding a deep cut

added by our algorithm for semidefinite programming are deep cuts, as illustrated in figure 2. The cuts added by the algorithm have the form $d^T(C - \mathcal{A}^T y)d \geq 0$.

Note that $S \succeq 0$ is equivalent to $\lambda_{min}(S) \geq 0$, so (SDD) can also be written as

$$\begin{array}{lrcl} \max & b^T y & & \\ \text{s.t.} & \lambda_{min}(C - \mathcal{A}^T y) & \geq & 0 \end{array} \tag{8}$$

We have a convex programming formulation in (8): we are maximizing a linear function subject to a convex constraint $\lambda_{min}(C - \mathcal{A}^T y) \geq 0$. The minimum eigenvalue function is a concave non-smooth function, with a discontinuous gradient whenever this eigenvalue has a multiplicity greater than one. Any supergradient to the minimum eigenvalue function at a point $\bar{y}$ is given by $pp^T \bullet \mathcal{A}^T$, where p is a normalized eigenvector corresponding to $\lambda_{min}(C - \mathcal{A}^T \bar{y})$. Thus we have

$$\lambda_{min}(C - \mathcal{A}^T y) \;\leq\; \lambda_{min}(C - \mathcal{A}^T \bar{y}) + pp^T \bullet (C - \mathcal{A}^T y - C + \mathcal{A}^T \bar{y}), \;\; \forall y \tag{9}$$

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

$$\lambda_{min}(C - \mathcal{A}^T \bar{y}) + pp^T \bullet (C - \mathcal{A}^T y - C + \mathcal{A}^T \bar{y}) \;\geq\; 0 \tag{10}$$

To motivate this consider (10) with the reversed inequality. This would imply $\lambda_{min}(C - \mathcal{A}^T y) < 0$ from (9), and so y also violates the convex constraint. It follows that any feasible y satisfies (10). Using the fact that p is a normalized eigenvector corresponding to $\lambda_{min}(C - \mathcal{A}^T \bar{y})$, we have

$$\begin{aligned} pp^T \bullet (C - \mathcal{A}^T \bar{y}) &= p^T (C - \mathcal{A}^T \bar{y}) p \\ &= \lambda_{min}(C - \mathcal{A}^T \bar{y}) \end{aligned}$$

Thus we can rewrite (10) as

$$pp^T \bullet (C - \mathcal{A}^T y) \geq 0 \tag{11}$$

a valid cutting plane which is satisfied by all the feasible y. Also, since $\lambda_{min}(C - \mathcal{A}^T \bar{y}) < 0$, this is a *deep* cutting plane.

Another characterization of $S \succeq 0 \in \mathcal{S}^n$ is $\lambda_i \geq 0,\ i = 1, \ldots, n$. Thus any eigenvector corresponding to a negative eigenvalue of S gives a valid cutting plane.

Since the most negative eigenvalue of a symmetric matrix and its associated eigenvector can be computed in polynomial time, the SDP actually has a polynomial time separation oracle. We can thus use this oracle in conjunction with the ellipsoid algorithm to generate a polynomial time algorithm for the SDP (this follows from the equivalence of separation and optimization established in Grotschel et al [7]; more details can be found in section 5).

The variational characterization of the minimum eigenvalue lets us restate the problem of finding the minimum eigenvalue as a quadratic optimization problem in d:

$$\min \quad \{d^T (C - \mathcal{A}^T \bar{y}) d : ||d||_2 \leq 1\} \tag{12}$$

Note that the objective function in (12) is a non-convex quadratic function. Thus any solution to (12) should occur on the boundary of the trust region $\{d : ||d||_2 = 1\}$. Thus we can replace $\{d : ||d||_2 = 1\}$ with $\{d : ||d||_2 \leq 1\}$. This suggests a generalization where d is drawn from a different region, so

$$\min \quad \{d^T (C - \mathcal{A}^T \bar{y}) d : d \in B\} \tag{13}$$

for some compact set B containing the origin. This subproblem involves minimizing a non-convex objective function, and is usually an *NP hard* problem. However, in the 2 norm case, we have a polynomial time routine thanks to the variational characterization of the minimum eigenvalue function.

In particular, we can use the infinity norm, which finds a cutting plane by solving the subproblem

$$\min \quad \{d^T (C - \mathcal{A}^T \bar{y}) d : ||d||_\infty \leq 1\}. \tag{14}$$

This approach has advantages when solving the semidefinite programming relaxations of combinatorial optimization problems, because the extreme points of this region are ± 1 vectors, which correspond to solutions in the underlying combinatorial optimization problem. The disadvantage is that the subproblem is NP-hard, so we search for a local minimum rather than a global minimum. For details, see [18].

In the semi-infinite literature, the scheme we described above is referred to as an *exchange method* (see the survey paper by Hettich and Kortanek [12]: section 7.1). The notation *exchange algorithm* refers to the fact that in every step a number of constraints are added (Step 2), and some of the old constraints may conceivably be deleted, i.e. an exchange of constraints takes place. This is similar to column generation approaches utilized in mixed-integer programming, except that we are dealing with an infinite, rather than a combinatorial, number of constraints.

The basic interior point algorithm for the SDP is the following:

1. **Initialize:** The feasible region $S \succeq 0$ is unbounded. To implement any cutting plane method we need a bounded feasible set. The idea is to enclose the feasible region in a large box, whose dimensions are determined by the tolerance to which we wish to solve the SDP relaxation. This is not entirely trivial. This gives our starting LP relaxation. For the maxcut problem, we have an easy starting relaxation (not bounded) thanks to the problem structure.

2. **Restart** the current LP relaxation with a strictly feasible starting point.

3. **Solve the LP relaxations** cheaply, to a moderate tolerance β on the relative duality gap. The solution to the dual LP provides an upper bound on the SDP value. Reduce β.

4. **Call the SDP separation oracle** to find violated cutting planes. Bucket sort the violated inequalities by violation, and add a subset of constraints to the relaxation. Drop any constraints that no longer appear important.

5. **Update lower bound:** In each iteration S is not quite psd. Find a feasible $S^{\#} \succeq 0$ by perturbing S slightly so that it is just psd, and update the lower bound.

6. **Check for termination:** If the difference between the upper and lower bounds is small, and the magnitude of the most negative eigenvalue of S is small, STOP.

7. **Loop:** return to step 2

For more details on the steps of this algorithm see [15, 16, 17, 18]. More information on the implementation of interior point cutting plane algorithms can be found in [20].

Updating the lower bounds in step 5 can be done quickly if there exists a vector $\hat{y}$ satisfying $\mathcal{A}^T\hat{y} = I$. (For more on $\hat{y}$ see assumption 3 and lemma 2 in §6.) In particular, we can subtract a multiple of $\hat{y}$ from the current iterate in order to obtain a vector y for which the dual slack matrix $S = C - \mathcal{A}^T y$ is positive semidefinite. The lower bound differs from the current dual objective value by the product of this multiple and $b^T\hat{y}$. One appropriate multiple is to take the size of the most negative eigenvalue of the current dual slack matrix. It follows that if S is close to being psd then the duality gap for (SDP) and (SDD) in step 6 is small. For more sophisticated update schemes, see [15, 16, 17, 18].

We illustrate one iteration of the cutting plane approach in figure (3). Each iteration in the interior point cutting plane approach contains the following three ingredients:

1. **Restart with a strictly feasible starting point:** The oracle in the previous iteration returned a number of deep cuts, and the current iterate is no longer feasible. We first work towards restoring strict feasibility of the primal and dual iterates, using extensions of techniques given in [22].

2. **Solve the LP relaxations cheaply using an interior point solver:** We solve the LP relaxations, with the strictly feasible starting point from step 1, approximately, gradually tightening the accuracy as needed. This generates better cutting planes, because the cuts are generated at points that are more central.

3. **Call the oracle to generate deep cutting planes:** We experimented with two different oracles generating deep cutting planes, based on the 2-norm and the ∞-norm.

5 The complexity of the cutting plane approach

It must be mentioned that the SDP 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 the SDP mentioned in the previous section in conjunction with the ellipsoid algorithm to actually solve the SDP in polynomial time. It is interesting to compare the worst case complexity of such a method, with that of interior point methods.

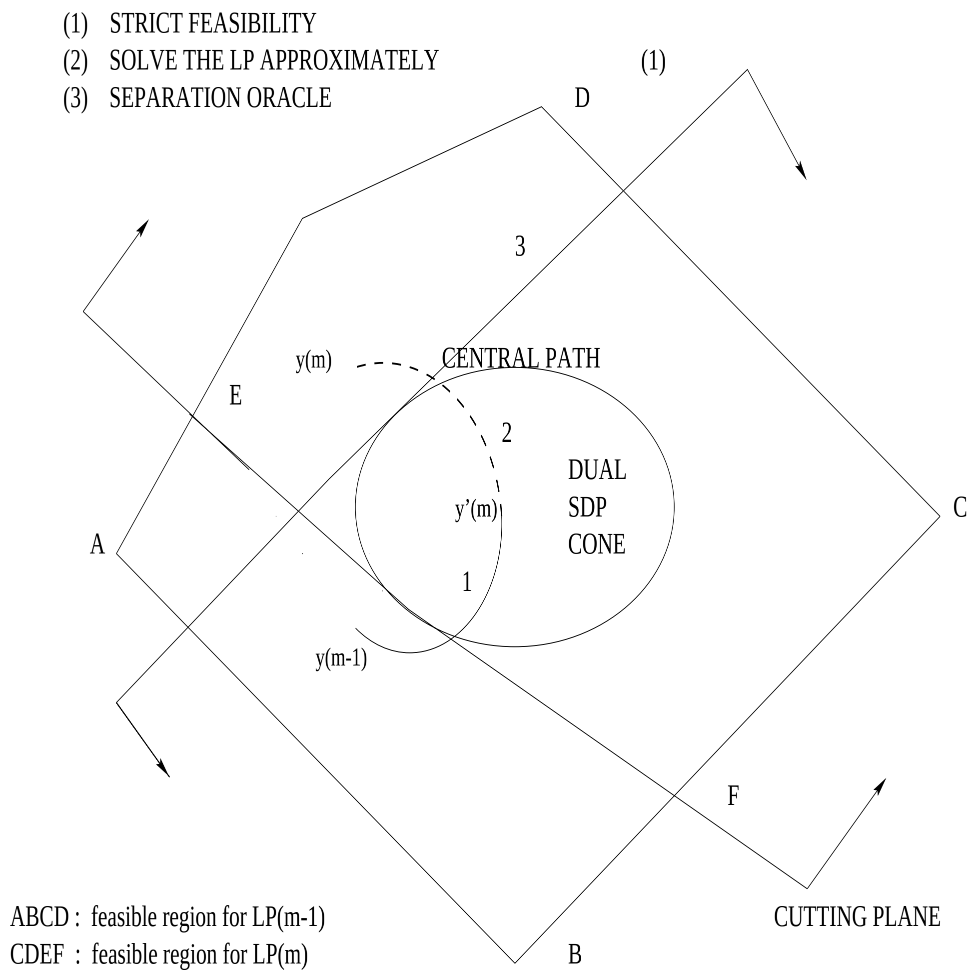

Figure 3: Solving the SDP via an LP cutting plane scheme: The three important stages in every iteration. The point $y(m-1)$ is cut off by the cutting plane EF. A strictly feasible restart point $y'(m)$ is found. The new linear program with feasible region $CDEF$ is solved approximately, giving $y(m)$. The separation oracle is then used to find a cutting plane violated by $y(m)$.

The ellipsoid algorithm (for example, see Grotschel et al [7]) can solve a convex programming problem of size k with a separation oracle to an accuracy of ϵ, in $O(k^2 \log(\frac{1}{\epsilon}))$ 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 [32] and refined by Anstreicher [2, 3] and Ramaswamy and Mitchell [29]. (See [21] 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 [23].

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, $i = 1, \ldots, k$ are in $\mathcal{S}^n$. This can be done in $O(kn^2)$ arithmetic operations.

2. We then compute $\lambda_{min}(\bar{S})$, and an associated eigenvector $\bar{d}$. 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 $\sum_{i=1}^k y_i(\bar{d}^T A_i \bar{d}) \leq \bar{d}^T C \bar{d}$, 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 6 *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 Todd [30] for more details) can solve an SDP of size n, to a precision ϵ, 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 have special structures (such as rank one as in the maxcut problem).

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 6. In particular, the problem $(SDPN)$ is in exactly the form of (SDD), so the cutting plane approach of §4 can be applied to it directly. It follows from theorem 6 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 [24]. 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.

6 The dimension of the faces generated in the y-space

6.1 The tangent space

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. 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 Γ be a nonempty convex set in $\mathbb{R}^n$. The closure of Γ is denoted $cl(\Gamma)$. For any point $u \in \mathbb{R}^n$, the distance from u to Γ is defined as the distance from u to the unique closest point in $cl(\Gamma)$ and is denoted $dist(u, \Gamma)$. Let v be a point in the closure of Γ. The* cone of feasible directions*, the* tangent cone*, and the* tangent space *at v are defined as*

$$\begin{aligned} dir(v,\Gamma) &= \{d : v + td \in \Gamma \text{ for some } t > 0\} \\ tcone(v,\Gamma) &= cl(dir(v,\Gamma)) \\ &= \{d : dist(v + td, \Gamma) = o(t)\} \\ tan(v,\Gamma) &= \{d : dist(v \pm td, \Gamma) = o(t)\}. \end{aligned}$$

Given a supporting hyperplane H for Γ, the tangent space defined by H *at v is*

$$tpl(v,\Gamma,H) = \{d \in tcone(v,\Gamma) : v + d \in H\}.$$

A convex subset F of Γ is a face *of Γ if*

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

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

The equivalence of the two definitions of tcone follows from page 135 of Hiriart-Urruty and Lemaréchal [13]. The geometry of semidefinite programming is surveyed by Pataki [27]. The tangent space is a subset of the cone of tangents. Note that our definition of the tangent space defined by a hyperplane is a one-sided definition, that

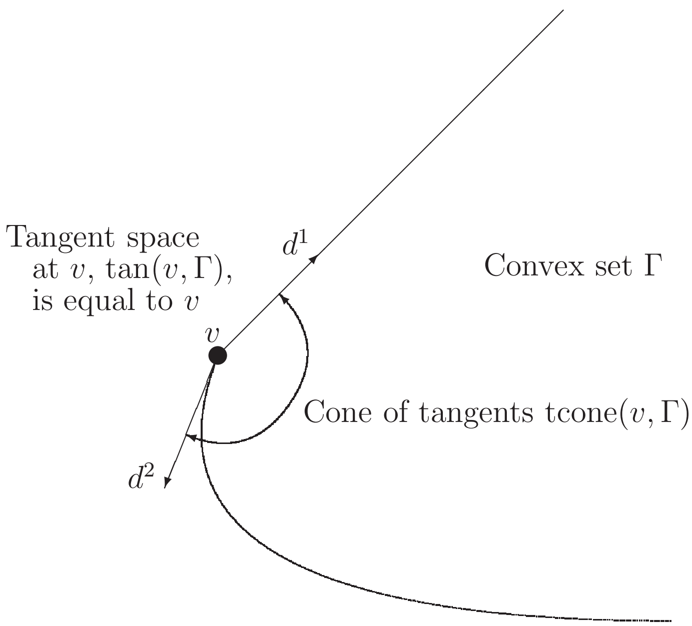

Figure 4: The tangent cone and tangent space for example 1

is, it depends on $d \in \text{tcone}(v, \Gamma)$ and not on $d \in \tan(v, \Gamma)$. 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.

Example 1 *Take $\Gamma \subseteq I\!R^2$ to be the vectors that make the following matrix S positive semidefinite:*

$$S = \begin{bmatrix} y_1 & y_2 & 0 \\ y_2 & y_1 - 3 & 0 \\ 0 & 0 & 2 - y_2 \end{bmatrix}.$$

Let $v = [4 \;\; 2]^T \in \Gamma$. This example is illustrated in figure 4.

The tangent space $\tan(v, \Gamma)$ consists of just the point v. The extreme rays of the cone of tangents are the vectors $d^1 = [1 \quad 0]^T$ and $d^2 = [-4 \quad -5]^T$. The ray d^2 corresponds to the linear constraint

$$5y_1 - 4y_2 \geq 12,$$

and the tangent space defined by this constraint at v is the halfline generated by d^2. It *is desirable to add constraints corresponding to higher dimensional faces of the cone of tangents, rather than weaker constraints that are active at v, such as $y_1 - 2y_2 \geq 0$.*

The set Y of feasible solutions y to (SDD) is a convex set, and the cutting planes we generate are valid inequalities. 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.

The tangent space to the cone $\mathcal{S}_+^n$ of $n \times n$ positive semidefinite matrices at a psd matrix X of rank r is characterized in the following theorem, given in Alizadeh et al [1]. We first define

$$\Xi(X) := \{Q \begin{bmatrix} U & V \\ V^T & 0 \end{bmatrix} Q^T : U \in \mathcal{S}^r, V \in I\!R^{r\times(n-r)}\} \tag{15}$$

where Q is the orthogonal matrix of the eigenvectors of X with the columns ordered so that a basis for the nullspace of X is given by the last $(n-r)$ columns of Q.

Theorem 7 *[1] Let X be a psd matrix of rank r. The tangent space to $\mathcal{S}_+^n$ at X is $\Xi(X)$.*

We now argue why any matrix in $\Xi(X)$ is in the tangent space. To see this, consider sufficiently small perturbations $X \pm \epsilon\Delta X$, with $\Delta X \in \Xi(X)$. These matrices are typically indefinite, but a psd matrix can be recovered by adding a matrix W of the form $W = Q \begin{pmatrix} 0 & 0 \\ 0 & \Lambda \end{pmatrix} Q^T$ to them, where $\Lambda \in \mathcal{S}_+^{n-r}$. Thus, the distance from $X \pm \epsilon\Delta X$ to $\mathcal{S}_+^n$ is the norm of the matrix Λ. We utilize the Schur complement idea to obtain a bound on Λ. Notice that

$$X + \epsilon\Delta X + W \quad = \quad Q \begin{bmatrix} \mathrm{Diag}(\lambda_1, \ldots, \lambda_r) + \epsilon U & \epsilon V \\ \epsilon V^T & \Lambda \end{bmatrix} Q^T$$

so we obtain the following condition on Λ:

$$\begin{aligned} X + \epsilon\Delta X + W \succeq 0 \quad &\Leftrightarrow \quad \begin{bmatrix} \mathrm{Diag}(\lambda_1, \ldots, \lambda_r) + \epsilon U & \epsilon V \\ \epsilon V^T & \Lambda \end{bmatrix} \succeq 0 \\ &\Leftrightarrow \quad \Lambda - \epsilon^2 V^T(\mathrm{Diag}(\lambda_1, \ldots, \lambda_r) + \epsilon U)^{-1} V \succeq 0 \end{aligned}$$

We have utilized the fact that $\mathrm{Diag}(\lambda_1, \ldots, \lambda_r)$ (since it is positive definite) dominates ϵU, so the matrix $\mathrm{Diag}(\lambda_1, \ldots, \lambda_r) + \epsilon U$ is invertible in the Schur complement. Thus loosely speaking we can choose Λ with $||\Lambda|| = O(\epsilon^2)$, so ΔX is in $\tan(X, \mathcal{S}_+^n)$. For a sufficiently small perturbation ϵ we can say that whenever $\Delta X \in \Xi(X)$, then the matrix $X \pm \epsilon\Delta X$ is sufficiently close to being a psd matrix.

With the 2-norm oracle, the cutting plane algorithm generates constraints of the form $d^T(\mathcal{A}^T y)d \leq d^T Cd$, where d is an eigenvector of the current dual slack matrix $\bar{S}$ with a negative eigenvalue. This constraint generates a (possibly empty) tangent space of the set Y of feasible solutions for (SDD). To simplify the discussion, it is helpful if Y is full-dimensional. Therefore we make the following assumption:

Assumption 3 *There exists a constant $a \geq 0$ such that every X satisfying $\mathcal{A}X = b$ also satisfies $tr(X) = a$.*

Helmberg [10] shows that every SDP whose primal feasible set is bounded can be rewritten to satisfy this assumption. The following lemma is a consequence of this assumption.

Lemma 2 *There exists a unique vector $\hat{y}$ satisfying $\mathcal{A}^T\hat{y} = I$, the identity matrix.*

Proof: The constraint $I \bullet X = a$ 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 2, the linear independence of the matrices A_i. $\square$

Subtracting a positive multiple of $\hat{y}$ from any $y \in Y$ gives a strictly feasible point, so Y is full-dimensional.

6.2 Satisfying the added constraint at equality

We show first that there is a point in Y that satisfies the cutting plane at equality, if the plane is chosen appropriately. Let $\bar{y}$ be the current dual iterate and let the dual slack matrix be $\bar{S} = C - \mathcal{A}^T\bar{y}$. Let $\bar{d}$ be an eigenvector of $\bar{S}$ of norm one corresponding to the most negative eigenvalue of $\bar{S}$, and let λ denote this eigenvalue. Define

$$\check{y} := \bar{y} - \lambda\hat{y}. \tag{16}$$

The dual slack matrix given by $\check{y}$ is $\check{S} = \bar{S} + \lambda I$, which is positive semidefinite, so $\check{y} \in Y$. Further, $\bar{d}$ is in the nullspace of $\check{S}$.

Theorem 8 *The constraint $\bar{d}^T(C - \mathcal{A}^T y)\bar{d} \geq 0$ is satisfied at equality by $\check{y}$.*

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

6.3 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 9 *If the minimum eigenvalue of $\bar{S}$ has multiplicity of one, then the constraint $\bar{d}^T(C-\mathcal{A}^T y)\bar{d} \geq 0$ defines a tangent space of Y of dimension $k-1$ at $\check{y}$.*

Proof: For any point y, define the direction $d = y - \check{y}$. The constraint can be written $\bar{d}^T(\check{S}-\mathcal{A}^T d)\bar{d} \geq 0$, or equivalently as

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

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

$$\sum_{i=1}^{k} d_i \bar{d}^T A_i \bar{d} = 0, \tag{17}$$

so $H = \{\check{y} + d : d \text{ satisfies (17)}\}$. We want to show that any d satisfying (17) is in $\tan(\check{y}, Y)$.

The matrix $\check{S}$ has an eigendecomposition

$$\check{S} = [Q_1 \; \bar{d}] \begin{bmatrix} \Lambda & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} {Q_1}^T \\ \bar{d}^T \end{bmatrix} \tag{18}$$

where $Q = [Q_1 \; \bar{d}]$ is an orthogonal matrix and Λ is a $(n-1)\times(n-1)$ positive diagonal matrix. For any solution to (17), the slack matrix at $\check{y}+\epsilon d$ in the transformed space Q can be written as

$$\begin{aligned} \mathring{S} &= \mathring{\check{S}} - \epsilon \mathring{\mathcal{A}^T} d \\ &= \begin{bmatrix} \Lambda + \epsilon U & \epsilon w^T \\ \epsilon w & 0 \end{bmatrix} \end{aligned}$$

where the norms of $U = (\mathring{\mathcal{A}^T} d)_{11}$ and $w = (\mathring{\mathcal{A}^T} d)_{12}$ are comparable to the norm of d, and the last line follows from (17). This matrix may not be positive semidefinite. However, there exists $\beta = O(\epsilon^2)$ such that adding $\beta\hat{y}$ to the point $\check{y}+\epsilon d$ gives a point with a psd slack matrix. It follows from definition 1 that any direction satisfying (17) is in $\tan(\check{y}, Y)$, so $\text{tpl}(\check{y}, Y, H)$ has dimension $k-1$. $\square$

For an illustration of this result, see figure 5.

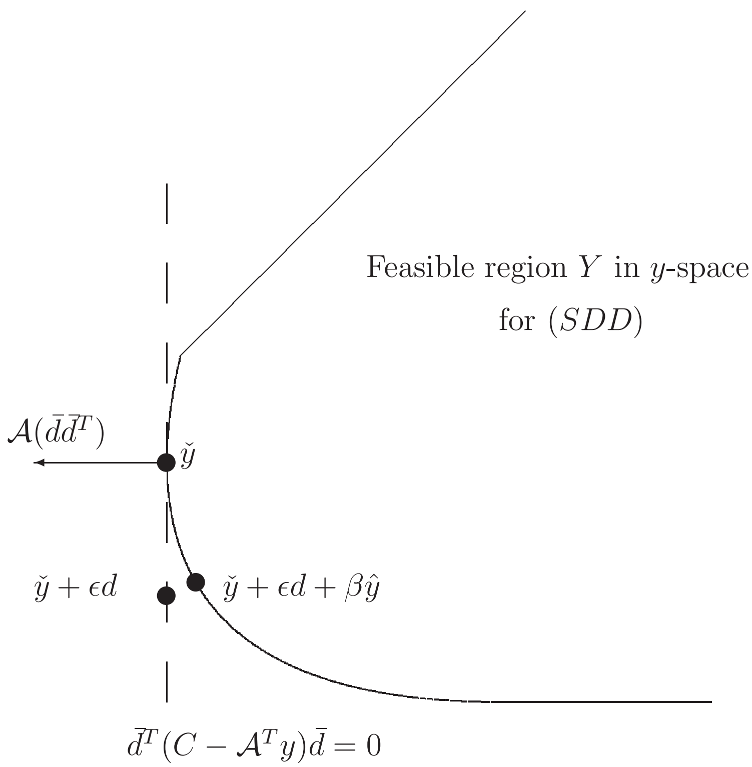

Figure 5: The tangent plane defined by the cutting plane in y-space

6.4 General values for the nullity

We can generalize theorem 9 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. We let r denote the nullity of $\check{S}$, and $\bar{d}$ denotes any vector of norm one in the nullspace of $\check{S}$. The generalization uses the eigendecomposition of $\check{S}$, given by:

$$\check{S} = [Q_1\ Q_2] \begin{bmatrix} \Lambda & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} {Q_1}^T \\ {Q_2}^T \end{bmatrix}, \tag{19}$$

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

Lemma 3 *If d is a unit vector such that the matrix $Q_2^T(\mathcal{A}^T d)Q_2$ is negative semidefinite then d is in the cone of tangents of Y at $\check{y}$.*

Proof: For any β, the dual slack matrix at the point $\check{y}+\epsilon d-\beta\hat{y}$ in the transformed space Q is

$$\begin{aligned} \mathring{S} &= \mathring{\check{S}} - \epsilon\mathring{\mathcal{A}}^T d + \beta I \\ &= \begin{bmatrix} \Lambda + \epsilon U + \beta I & \epsilon W^T \\ \epsilon W & \epsilon Z + \beta I \end{bmatrix} \end{aligned}$$

where the norms of $U = -(\mathring{\mathcal{A}}^T d)_{11}$, $W = -(\mathring{\mathcal{A}}^T d)_{12}$, and $Z = -(\mathring{\mathcal{A}}^T d)_{22}$ are comparable to the norm of d. By assumption, Z is positive semidefinite. This matrix is positive semidefinite if and only if the Schur complement

$$\epsilon Z + \beta I - \epsilon^2 W(\Lambda + \epsilon U + \beta I)^{-1} W^T$$

is psd. Thus, we can take β to be $O(\epsilon^2)$ and obtain a psd slack matrix, so by definition 1 we can conclude that d is in the cone of tangents to Y at $\check{y}$. □

The matrix $Q_2^T(\mathcal{A}^T d)Q_2 = 0$ is negative semidefinite if it is zero. This lets us place a lower bound on the dimension of the tangent space.

Theorem 10 *The dimension of the tangent space to Y at $\check{y}$ defined by the valid constraint $\bar{d}^T(C - \mathcal{A}^T y)\bar{d} \geq 0$ is at least $k - \begin{pmatrix} r+1 \\ 2 \end{pmatrix}$.*

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

$$p_i^T(\mathcal{A}^T d)p_j = 0, \qquad 1 \leq i \leq j \leq r. \tag{20}$$

Note that $\bar{d}$ is a linear combination of the columns of Q_2, so any d satisfying (20) also satisfies $\bar{d}^T(\mathcal{A}^T d)\bar{d} = 0$. From lemma 3, any direction satisfying (20) is in the cone of tangents to Y at $\check{y}$. The result follows. $\square$

Note from theorem 4 that this result can be quite weak. In §7, we give examples where the tangent space has dimension far greater than implied by theorem 10. It is desirable to choose the vector $\bar{d}$ to make the dimension of the corresponding tangent plane as large as possible. We now characterize the vectors $\bar{d}$ which will do this. First, we note that a converse to lemma 3 holds, enabling us to characterize the cone of tangents tcone$(\check{y}, Y)$.

Lemma 4 *If $d \in tcone(\check{y}, Y)$ then $Q_2^T(\mathcal{A}^T d)Q_2$ is negative semidefinite.*

Proof: Assume $Q_2^T(\mathcal{A}^T d)Q_2$ has a positive eigenvalue with corresponding unit eigenvector v. Note that we must have $v = Q_2^T u$ for some vector u. Without loss of generality, we can assume u is in the nullspace of Q_1^T. By a similar argument to that in the proof of lemma 3, the slack matrix at the point $\check{y} + \epsilon d$ in the transformed space Q is

$$\mathring{S} = \begin{bmatrix} \Lambda + \epsilon U & \epsilon W^T \\ \epsilon W & \epsilon Z \end{bmatrix}$$

where $U = -(\mathring{\mathcal{A}}^T d)_{11}$, $W = -(\mathring{\mathcal{A}}^T d)_{12}$, and $Z = -(\mathring{\mathcal{A}}^T d)_{22}$. It follows that $u^T S u = \epsilon v^T Z v$, which is negative and $O(\epsilon)$. Thus, d is not in the closure of the set of feasible directions. $\square$

Taken together, lemmas 3 and 4 imply the following:

Theorem 11 *The cone of tangents at $\check{y}$ in Y is the set of directions d for which the matrix $Q_2^T(\mathcal{A}^T d)Q_2$ is negative semidefinite.*

We denote this cone by C, so

$$\begin{aligned} C &:= \{d : Q_2^T(\mathcal{A}^T d)Q_2 \preceq 0\} && (21) \\ &= \{d : \sum_{i=1}^{k} Q_2^T A_i Q_2 d_i \preceq 0\}. && (22) \end{aligned}$$

We define the dual cone to C to be the set of all vectors v satisfying $v^T d \leq 0$ for all $d \in C$. This is given by

$$\begin{aligned} C^* &= \mathrm{cl}(\{\mathcal{A}(Q_2 X Q_2^T) : X \succeq 0\}) && (23) \\ &= \mathrm{cl}(\{\mathcal{A}(uu^T) : u \text{ is in the null space of } \check{S}\}). && (24) \end{aligned}$$

It should be noted that the cone $\{\mathcal{A}(Q_2 X Q_2^T) : X \succeq 0\}$ may not be closed. For conditions to guarantee the closure of this cone, see Pataki [28]. An example drawn from this reference that results in this cone not being closed is as follows:

Example 2 *[28] Take $k = n = 2$. Let*

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

The ray $(0,1)^T$ is not in the cone $\{\mathcal{A}(Q_2 X Q_2^T) : X \succeq 0\}$, but it is in its closure. For this example, C is the set of vectors of the form $(d_1, 0)$ with $d_1 \leq 0$ and C^* is all vectors (d_1, d_2) with $d_1 \geq 0$. It should be noted that the matrices in this example do not satisfy assumption 3.

By the definition of a dual cone, any vector $v \in C^*$ gives a valid constraint $v^T y \leq v^T \check{y}$ for Y. The strongest constraints for C are those where v is an extreme ray of C^*. If C^* is closed, then we can find extreme rays of C^* by finding extreme points of slices through C^*. This is the case if assumption 3 holds, as we show below in theorem 13. Define

$$\Pi := \{v \in C^* : \hat{y}^T v = 1\}. \qquad (25)$$

We show that Π is a bounded set and that C^* is a homogenization of Π. First, we give a useful lemma.

Lemma 5 *For any $v \in C^*$ with $v = \mathcal{A}(Q_2 X Q_2^T)$ and $X \succeq 0$, the quantities $\hat{y}^T v$ and $tr(X)$ are equal.*

Proof: We have

$$\begin{aligned}
\hat{y}^T v &= \hat{y}^T \mathcal{A}(Q_2 X Q_2^T) \\
&= \mathrm{tr}((\mathcal{A}^T \hat{y})(Q_2 X Q_2^T)) \\
&= \mathrm{tr}(Q_2 X Q_2^T) \qquad \text{from assumption 3} \\
&= \mathrm{tr}(Q_2^T Q_2 X) \\
&= \mathrm{tr}(X).
\end{aligned}$$

$\square$

Theorem 12 *The set Π is closed and bounded.*

Proof: From lemma 5, we have $\Pi = \{\mathcal{A}(Q_2 X Q_2^T) : \mathrm{tr}(X) = 1, X \succeq 0\}$. Thus, Π is the image under a linear transformation of the closed and bounded set $\{X \succeq 0 : \mathrm{tr}(X) = 1\}$, so Π must also be closed and bounded. $\square$

Theorem 13 *The cone C^* is closed, with $C^* = \{v : v = \lambda w, \lambda \geq 0, w \in \Pi\}$.*

Proof: It follows from lemma 5 that $C^* \subseteq \{v : \hat{y}^T v \geq 0\}$ and for any nonzero $v \in C^*$, we have $\hat{y}^T v > 0$, so $C^* = \{v : v = \lambda w, \lambda \geq 0, w \in \Pi\}$. Finally, the closedness of C^* follows from the boundedness of Π. $\square$

It follows from theorems 12 and 13 that we can find all the extreme rays of C^* 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 C^*. This SDP is easy to solve, requiring just the calculation of the minimum eigenvalue of an $r \times r$ matrix.

Theorem 14 *The extreme rays of C^* 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 $(Q_2^T \mathcal{A} Q_2)^T g$ for some vector g.*

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

$$\begin{array}{llcrcl}
\min & & & g^T v & & \\
\text{subject to} & \mathcal{A}(Q_2 X Q_2^T) & - & v & = & 0 \\
& & & \hat{y}^T v & = & 1 \\
& X & & & \succeq & 0
\end{array} \qquad (SP14)$$

for some vector g. The dual of this problem is

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

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

$$\begin{array}{lrcl} \max & z & & \\ \text{subject to} & zI & \preceq & (Q_2^T \mathcal{A} Q_2)^T g. \end{array} \qquad (SD14)$$

It follows that the optimal value z is the smallest eigenvalue of $(Q_2^T \mathcal{A} Q_2)^T g$. 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 15 *Let v be an extreme ray found by solving $(SP14)$, and assume the nullity of the optimal slack matrix in $(SD14)$ is one. The tangent space defined by the constraint $v^T y \leq v^T \breve{y}$ has dimension at least $k - r$.*

Proof: Let u be a basis for the null space of the optimal slack matrix for $(SD14)$. By complementary slackness for the SDP $(SP14)$, 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 $(SP14)$ and let $\tilde{d} = \hat{y}z - g$. From complementary slackness in the SDP pair in theorem 14 we have $v^T \tilde{d} = 0$. Also $Q_2^T \mathcal{A}^T(\tilde{d}) Q_2 \preceq 0$. Hence it is clear that $\tilde{d}$ is in the tangent space defined by the constraint.

We now exploit the hypothesis on the nullity of $Q_2^T \mathcal{A}^T(\tilde{d}) Q_2$. Without loss of generality we order its eigenspace such that the first $r - 1$ eigenvalues are strictly negative, and the last eigenvalue is zero. Moreover u is the eigenvector corresponding to this zero eigenvalue. 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 consider $Q_2^T \mathcal{A}^T(d)Q_2$ in the transformed space of the eigenvectors of $Q_2^T \mathcal{A}^T(d')Q_2$.

$$Q_2^T \mathcal{A}^{\overset{\circ}{T}}(d)Q_2 \quad = \quad \begin{bmatrix} \mathrm{Diag}(\lambda) + \alpha V & 0 \\ 0 & 0 \end{bmatrix}$$

Since $\mathrm{Diag}(\lambda) \prec 0$, 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 $d' \in I\!R^k$ 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 3 holds. It follows that the minimum eigenvalue will have multiplicity one almost surely.

7 Examples of the tangent space

In this section, we look at examples of the tangent spaces for some semidefinite programs. The positive orthant $I\!R^n_+$ can be expressed as the feasible region of a semidefinite program. This simple SDP is the subject of §7.1. A more complicated example with a large value of r is the subject of §7.2.

7.1 First example

Take

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

The dual slack matrix is $S = \mathrm{Diag}(y)$. This gives $Y = I\!R^n_+$, and $k = n$. Thus, Y is polyhedral and so tangent spaces correspond to faces. Assumption 3 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 (16) is a nonnegative vector with r components equal to zero. The corresponding $\check{S}$ is a diagonal matrix. The vectors $\bar{d}$ in the nullspace of $\check{S}$ are those satisfying $\bar{d}_i \check{y}_i = 0$ for each component i. The constraint given in (11) is then $\sum_{i=1}^n \bar{d}_i^2 y_i \geq 0$.

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

If several components $\check{y}_{n-r+1}, \ldots, \check{y}_n$ are zero, we can take $\bar{d} = e_i$, 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 $\bar{d} = \sum_{i=n-r+1}^n e_i$, giving the $k-r$ dimensional tangent space defined by the hyperplane $\sum_{i=n-r+1}^n y_i = 0$. Taking g in $(SP14)$ with $g_n < g_i$, $i = n-r+1, \ldots, n-1$ gives $v = -e_n$, leading again to the $k-1$ dimensional tangent space defined by the hyperplane $y_n = 0$.

7.2 Second example

Take

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

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 $\bar{d} = e_i - e_j$ 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 $\check{y}$. Then we can increase any other y_k and still stay on the tangent plane given by the hyperplane $d_i + d_j = 0$ and stay feasible. Also, we can move in the direction $e_i - e_j$ and stay on the hyperplane. If we move a distance ϵ in this direction, the 2×2 submatrix of S corresponding to components i and j has the form

$$\begin{bmatrix} 1+\epsilon & 1 \\ 1 & 1-\epsilon \end{bmatrix},$$

which has eigenvalues approximately $2 + 0.5\epsilon^2$ and $-0.5\epsilon^2$. If $n = 2$ then subtracting $0.5\epsilon^2 \hat{y}$ from y gives back a PSD dual slack matrix, so this direction is also in the tangent space, and the tangent space has dimension 1 $(= k - r)$. If $n > 2$ then the smallest eigenvalue of this modified dual slack matrix is negative and $O(\epsilon)$, so this direction is not in the tangent space: letting $v = e_i - 2e_j + e_l$ for some $l \neq i$, $l \neq j$, we get $v^T S v = -\epsilon$. Any other direction having $d_i + d_j = 0$ will also have the smallest eigenvalue of $S(\check{y} + \epsilon d)$ negative and at least $O(\epsilon)$. Thus, the dimension of the tangent space defined by the hyperplane $d_i + d_j = 0$ is $n - 2$, which is strictly larger than $k - r$ for $n \geq 4$.

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

$$\bar{d}_i = \begin{cases} \lfloor \frac{n}{2} \rfloor & \text{for } i = 1, \ldots, \lceil \frac{n}{2} \rceil \\ -\lceil \frac{n}{2} \rceil & \text{for } i = \lceil \frac{n}{2} \rceil + 1, \ldots, n \end{cases}$$

Since $n \geq 4$, at least two components of $\bar{d}$ 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 $\sum_{i=1}^{n} d_i \bar{d}_i^2 = 0$. In this situation, it can be shown that the dimension of the tangent space is zero, which is strictly smaller than $k - r$.

8 Adding nonpolyhedral cutting planes

We noted in §3 that (LPR) can be rewritten as

$$\begin{array}{lllll} \min & C \bullet (DMD^T) & & & \\ \text{s.t.} & A_j \bullet (DMD^T) & = & b_j & j = 1, \ldots, k \\ & M & \succeq & 0 & \\ & M & \text{diagonal} & & \end{array}$$

where the columns of the $n \times m$ matrix D are the cutting planes $d_i, i = 1, \ldots, m$. Relaxing the condition that M be diagonal to the condition that it be block diagonal corresponds to adding nonpolyhedral cutting planes. Let D^i be the columns of D corresponding to the ith diagonal block of M and let p denote the number of blocks. The dual problem is then

$$\begin{array}{llllll} \max & b^T y & & & & \\ \text{subject to} & {D^i}^T (\mathcal{A}^T y) D^i & + \; S^i & = & {D^i}^T C D^i & i = 1, \ldots, p \\ & & S^i & \succeq & 0 & i = 1, \ldots, p, \end{array} \qquad (SDDB)$$

as in Oskoorouchi and Goffin [25]. If each block consists of a single diagonal element then this reduces to (LDR). Each constraint of the form ${D^i}^T C D^i \succeq {D^i}^T (\mathcal{A}^T y) D^i$ can be added as a nonpolyhedral cutting plane. In this section, we consider the effect of such cutting planes on the dual feasible region.

Let $\bar{y}$ be the current dual iterate and let the dual slack matrix be $\bar{S} = C - \mathcal{A}^T \bar{y}$. Assume that this matrix is not positive semidefinite and let λ denote the magnitude of the most negative eigenvalue of $\bar{S}$. Let r denote the multiplicity of this eigenvalue. Let the columns of the $n \times r$ matrix Q_2 form a basis for the eigenspace corresponding to $-\lambda$. The constraint that is added to $(SDDB)$ is

$$Q_2^T (C - \mathcal{A}^T y) Q_2 \succeq 0. \qquad (26)$$

Defining $\check{y}$ as in (16), we have, analogous to theorem 8, that $Q_2^T(C - \mathcal{A}^T\check{y})Q_2 = 0$, since the columns of Q_2 give a basis for the nullspace of $C - \mathcal{A}^T\check{y}$.

It follows from theorem 11 that the set of feasible directions d at $\check{y}$ which satisfy constraint (26) is exactly the cone of tangents. Of course, the practical disadvantage of adding such a nonpolyhedral cutting plane lies in the increased complexity of calculating a new iterate. One way to reduce the complexity is to add a nonpolyhedral constraint that uses only a subset of the columns of Q_2, with the concomitant disadvantage that the cone of tangents is not captured as accurately.

9 Conclusions

The results of §5 show that cutting plane approaches to the solution of semidefinite programming problems have attractive theoretical complexity. Such an approach is more attractive for larger scale problems where standard interior point methods become impractical. Initial computational results are contained in [15, 16, 18], but more research is required.

The results of §6 show that a tangential cutting plane can always be found if the nullity of the dual slack matrix is one. Further, for higher values of the nullity, good cutting planes 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 plane, as we show in §8. Adding many cuts of this type over a number of iterations will lead to a large SDP, so some pruning is needed. It is also possible to combine new cutting planes with old ones in a nonpolyhedral manner; this is described in [17] and is also the motivation for the spectral bundle method [11].

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*, 22:63–89, 1997.

[3] K. M. Anstreicher. Towards a practical volumetric cutting plane method for convex programming. *SIAM Journal on Optimization*, 9:190–206, 1999.

[4] K. M. Anstreicher. The volumetric barrier for semidefinite programming. *Mathematics of Operations Research*, 25(3):365–380, 2000.

[5] S. Burer, R. D. C. Monteiro, and Y. Zhang. Solving a class of semidefinite programs via nonlinear programming. *Mathematical Programming*, 93(1):97–122, 2002.

[6] A. R. Conn, N. I. M. Gould, and P. L. Toint. *Trust Region Methods*. SIAM, Philadelphia, 2000.

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

[8] C. Helmberg. Semidefinite programming homepage, 1996. http://www-user.tu-chemnitz.de/~helmberg/semidef.html.

[9] C. Helmberg. SBmethod: a C++ implementation of the spectral bundle method. Technical Report 00–35, TU Berlin, Konrad-Zuse-Zentrum, Berlin, October 2000.

[10] C. Helmberg. Semidefinite programming for combinatorial optimization. Technical Report ZR-00-34, TU Berlin, Konrad-Zuse-Zentrum, Berlin, October 2000. Habilitationsschrift.

[11] C. Helmberg and F. Rendl. A spectral bundle method for semidefinite programming. *SIAM Journal on Optimization*, 10(3):673–696, 2000.

[12] R. Hettich and K. O. Kortanek. Semi-infinite programming: theory, methods and applications. *SIAM Review*, 35:380–429, 1993.

[13] J.-B. Hiriart-Urruty and C. Lemaréchal. *Convex analysis and minimization algorithms I*. Springer-Verlag, Berlin, 1993.

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

[15] K. Krishnan. *Linear programming approaches to semidefinite programming problems*. PhD thesis, Mathematical Sciences, Rensselaer Polytechnic Institute, Troy, NY 12180, July 2002.

[16] K. Krishnan and J. E. Mitchell. Semi-infinite linear programming approaches to semidefinite programming (SDP) problems. Technical report, Mathematical Sciences, Rensselaer Polytechnic Institute, Troy, NY 12180, August 2001. Accepted for publication in the Fields Institute Communications Series, Volume 37, "Novel approaches to hard discrete optimization problems", edited by P. Pardalos and H. Wolkowicz, pages 121–140, 2003.

[17] K. Krishnan and J. E. Mitchell. Cutting plane methods for semidefinite programming. Technical report, Mathematical Sciences, Rensselaer Polytechnic Institute, Troy, NY 12180, November 2002.

[18] K. Krishnan and J. E. Mitchell. Semidefinite cutting plane approaches for the maxcut problem. Technical report, Mathematical Sciences, Rensselaer Polytechnic Institute, Troy, NY 12180, May 2003.

[19] M. Laurent and R. Rendl. Semidefinite programming and integer programming. Technical report, CWI, Amsterdam, 2002.

[20] J. E. Mitchell. Computational experience with an interior point cutting plane algorithm. *SIAM Journal on Optimization*, 10(4):1212–1227, 2000.

[21] J. E. Mitchell. Polynomial interior point cutting plane methods. Technical report, Mathematical Sciences, Rensselaer Polytechnic Institute, Troy, NY 12180, July 2001. Revised April 2003. Accepted for publication in *Optimization Methods and Software*.

[22] J. E. Mitchell and M. J. Todd. Solving combinatorial optimization problems using Karmarkar's algorithm. *Mathematical Programming*, 56:245–284, 1992.

[23] A. S. Nemirovsky and D. B. Yudin. *Problem Complexity and Method Efficiency in Optimization*. John Wiley, 1983.

[24] Y. E. Nesterov and A. S. Nemirovsky. *Interior Point Polynomial Methods in Convex Programming : Theory and Algorithms*. SIAM Publications. SIAM, Philadelphia, USA, 1993.

[25] M. R. Oskoorouchi and J.-L. Goffin. The analytic center cutting plane method with semidefinite cuts. *SIAM Journal on Optimization*, 13(4):1029–1053, 2003.

[26] G. Pataki. On the rank of extreme matrices in semidefinite programs and the multiplicity of optimal eigenvalues. *Mathematics of Operations Research*, 23:339–358, 1998.

[27] 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, Dordrect, The Netherlands, 2000.

[28] G. Pataki. On the closedness of the linear image of a closed convex cone. Part I: necessary and sufficient conditions for nice cones, and core separation. Technical report, Department of Operations Research, University of North Carolina, Chapel Hill, 2003.

[29] 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.

[30] M. J. Todd. Semidefinite optimization. *Acta Numerica*, 10:515–560, 2001.

[31] M. J. Todd and Y. Ye. A centered projective algorithm for linear programming. *Mathematics of Operations Research*, 15(3):508–529, 1990.

[32] P. M. Vaidya. A new algorithm for minimizing convex functions over convex sets. *Mathematical Programming*, 73:291–341, 1996.

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

[34] H. Wolkowicz, R. Saigal, and L. Vandenberghe, editors. *Handbook of Semidefinite Programming: Theory, Algorithms, and Applications*. Kluwer Academic Publishers, Dordrecht, The Netherlands, 2000.

[35] 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.