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

Kartik Krishnan Sivaramakrishnan
Axioma, Inc.
Atlanta, GA 30350
kksivara@gmail.com
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 July 11, 2012 (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. The intersection of a cutting plane at a boundary point y with the tangent cone at that point gives a face of the tangent cone; we propose to use the dimension of this face to measure the strength of the cutting plane, we characterize the supporting hyperplanes that give higher dimensional faces of the tangent cone, 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. We present computational results that highlight the strength of these cutting planes in a practical setting. Our technique of finding higher dimensional cutting planes can conceivably be used to improve the convergence of the spectral bundle method of Helmberg et al. [10, 11], and the non-polyhedral cutting surface algorithms of Sivaramakrishnan et al. [38] and Oskoorouchi et al. [27, 28].

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

[1] 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 for solving semidefinite programs, and important applications of semidefinite programming in combinatorial optimization, control, robust optimization, and polynomial optimization (see Laurent and Rendl [19], for example).

Primal-dual interior point methods (IPMs) (see the surveys by de Klerk [7] and Monteiro [23]) 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 (see, for example, Toh et al. [40]). 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 been developed to solve large scale SDPs; these include: the low rank factorization approach of Burer and Monteiro [6]; the spectral bundle methods of Helmberg et al. [10, 11]) and Nayakkankuppam [24]; parallel implementations of primal-dual IPMs on shared memory (see Borchers and Young [5]) and distributed memory (see Yamashita et al. [42]) systems; and interior point cutting plane algorithms (see Sivaramakrishnan et al. [37, 38], Oskoorouchi et al. [27, 28], and Sherali and Fraticelli [36]).

In this paper, we investigate the properties of the interior point cutting algorithms presented in Krishnan et al. [15, 16, 38]. 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. Recently Qualizza et al. [33] have employed a variant of this algorithm to solve semidefinite relaxations of quadratically constrained quadratic problems. 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; this is the focus of section 5. 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 [25]), the strongest cutting planes are facet inequalities that describe the higher dimensional faces of the integer polytope. Each supporting hyperplane induces a face of the tangent cone. The dimension of this face is one measure of the strength of this hyperplane. When 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 face of the tangent cone 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 face of the tangent cone of dimension at least $k-r$. We illustrate these concepts with two representative examples in section 6, and some computational results can be found in section 7.

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 $\mathrm{trace}(AB)$.

2 Semi-infinite formulations for semidefinite programming

Consider the semidefinite programming problem

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

with dual

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

where $X, S \in \mathcal{S}^n_+$, $C \in \mathcal{S}^n$, b and y are vectors in $I\!R^k$, 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 *The matrices A_i, $i = 1, \ldots, k$ 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 trace$(X) = a$.*

Helmberg [9] 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 *[9] There exists a unique vector $\hat{y}$ satisfying $\mathcal{A}^T\hat{y} = I$, the identity matrix.*

Consider any feasible point y in (SDD). The point $(y - \mu\hat{y})$ is strictly feasible in (SDD) for $\mu > 0$. Indeed, the dual slack matrix at this new point is $S = (C - \mathcal{A}^T(y - \mu\hat{y})) = (C - \mathcal{A}^T y + \mu I) \succ 0$. 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

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

where B is a compact set, typically $\{\eta : ||\eta||_2 \leq 1\}$ or $\{\eta : ||\eta||_\infty \leq 1\}$. 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 $I\!R^{\frac{n(n+1)}{2}}$ variables.

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

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

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

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 nonzero vectors $\{\eta_i, i = 1, \ldots, m\}$, we obtain the following relaxation

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

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

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

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

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

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 [17]. In particular, in Krishnan and Mitchell [16] the relaxations are linear programs; in Sivaramakrishnan et al. [38] and Oskoorouchi and Goffin [27], 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 [11], 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

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

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]). □

The following theorem is due to Pataki [30] (also see Alizadeh et al. [1]):

Theorem 2 *There exists an optimal solution X^* with rank r satisfying the inequality $\frac{r(r+1)}{2} \le k$, where k is the number of 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 imply that X^* and S^* share a common eigenspace. Moreover, the positive eigenspace of X^* corresponds to the nullspace of S^* (see Alizadeh et al. [1]). Therefore, Theorem 2 also provides an upper bound on the dimension of the nullspace of S^*.

Let $q = \dfrac{n(n+1)}{2} - k$. 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 nullspace 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 $u \in I\!R^q$. The problem (SDP) can then be written equivalently as

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

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 section 4. (A similar nullspace representation of linear programming problems has been analyzed in the interior point literature; see, for example, Todd and Ye [39] and Zhang et al. [43].)

3 Cutting plane algorithms for semidefinite programming

Let

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

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 [13]). 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 [11]), using an analytic center with objective function cuts used to push the iterates towards an optimal solution (see Oskoorouchi and Goffin [27], for example), and only solving (LDR) approximately with an interior point method (see Sivaramakrishnan et al. [38], 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 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 [13]) is given by

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

where conv denotes the convex hull operation. There is also an alternate description

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

where $P \in I\!R^{n\times r}$ is an orthonormal matrix containing the eigenspace of $\lambda_{\max}(\mathcal{A}^T\bar{y} - C)$ (see Overton [29]). We note that (5) does not involve the convex hull operation. We will use the second expression (5) in Theorem 5 to derive an expression for the normal cone for the maximum eigenvalue function. Any subgradient from (4) gives the valid inequality

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

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

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

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

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

which is a valid cutting plane which is satisfied by all the feasible y. From linear algebra (see Horn and Johnson [14]), we have

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

Moreover, any eigenvector η corresponding to a positive eigenvalue of the matrix $(\mathcal{A}^T\bar{y} - C)$ gives a valid cutting plane $\eta^T(C - \mathcal{A}^Ty)\eta \geq 0$. 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 ∞-norm; for details, see Krishnan and Mitchell [18].

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 Sivaramakrishnan et al. [38]. Procedures to warm-start the new relaxations with strictly feasible starting points are discussed in Sivaramakrishnan et al. [38]. These are extensions of techniques given in Mitchell and Todd [22]. 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 Sivaramakrishnan et al. [38]. We illustrate two iterations of the cutting plane algorithm in figure 1.

Computational results with Algorithm 1 can be found in Sivaramakrishnan et al. [38]. Moreover, computational results obtained with the related spectral bundle and ACCPM algorithms can be found in Helmberg et al. [10, 11] and Oskoorouchi et al. [27, 28], 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 3 to a tolerance β. 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 β by a constant factor. Else, β 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, $S^m = (C - \mathcal{A}^T y^m)$ is not psd. Perturb S^m (using the vector $\hat{y}$ from Lemma 2) to generate $\check{y}$, that is on the boundary of Y and whose objective value $b^T \check{y}$ 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. [8]) can solve a convex programming problem of size k with a separation oracle to an accuracy of ϵ, in $O(k^2 \log(\frac{1}{\epsilon}))$

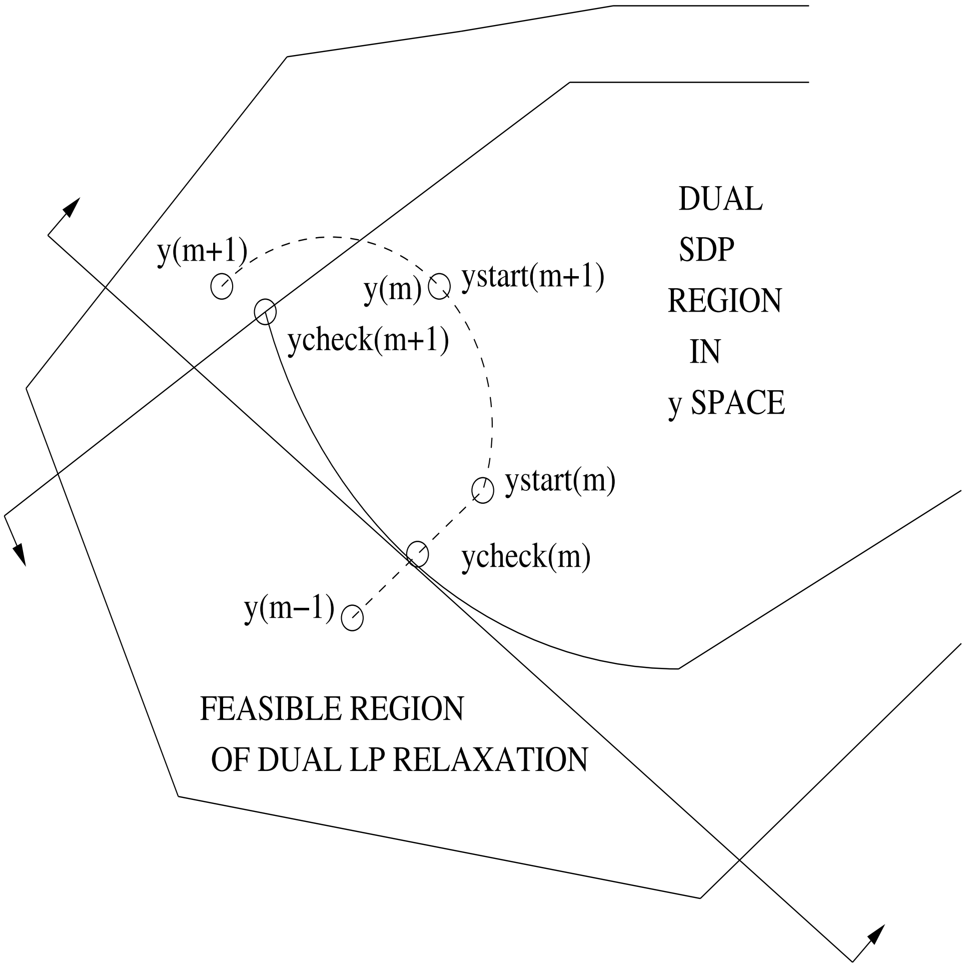

Figure 1: Solving the SDP via a cutting plane algorithm: In the $(m-1)$th iteration, the point $y(m-1)$ is outside the feasible region of (SDD). The oracle returns the unique hyperplane that supports the dual SDP feasible region at ycheck(m) and cuts off $y(m-1)$. A strictly feasible restart point ystart(m) 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, assuming the solution exists in a ball of radius $O(\frac{1}{\epsilon})$ centered at the origin. The number of arithmetic operations is $O(k^2 \log(\frac{1}{\epsilon})T + k^4 \log(\frac{1}{\epsilon}))$, 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, which accounts for the $k^4 \log(\frac{1}{\epsilon})$ term in the complexity measure.

The interior point cutting plane algorithm with the best complexity is a volumetric barrier method, due to Vaidya [41] and refined by Anstreicher [2] and Ramaswamy and Mitchell [34]. 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 [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 [26].

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 $\bar{y} \in I\!R^k$.

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 η. 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 still 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 [23] 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 are rank one as in the semidefinite relaxation of the maxcut problem (see Laurent and Rendl [19]).

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 $q = \dfrac{n(n+1)}{2} - k$ 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 $q = O(n)$.

5 Properties and generation of good cutting planes

The practical convergence properties of the interior point cutting plane algorithm could be strengthened by developing a better separation oracle. In this section, we 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 $\eta^T(\mathcal{A}^Ty)\eta \leq \eta^TC\eta$, where η is an eigenvector of the current dual slack matrix $\bar{S} = (C - \mathcal{A}^T\bar{y})$ with a negative eigenvalue. These constraints separate $\bar{y}$ from Y. We focus on the most negative eigenvalue. We show that the constraint corresponding to any eigenvector coming from the most negative eigenvalue is tight in Theorem 4. When the minimum eigenvalue has multiplicity one, we show that this constraint defines a facet of the tangent cone in Theorem 7. Furthermore, we construct a method for deciding between eigenvectors when the multiplicity is greater than one in section 5.3, culminating in the strong lower bound on the dimension of a cutting plane in Theorem 9.

We show first that there is a point $\check{y}$ on the boundary of Y that satisfies the cutting plane constraint at equality. Let η 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

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

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, η is in the nullspace of $\check{S}$.

Theorem 4 *Let η be an eigenvector of the current dual slack matrix $\bar{S}$ with minimum eigenvalue. The constraint $\eta^T(C - \mathcal{A}^T y)\eta \geq 0$ is satisfied at equality by the feasible point $\check{y}$.*

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

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 Λ is a $(n-r) \times (n-r)$ positive diagonal matrix. This eigendecomposition is used to characterize the cone of tangents at $\check{y}$ in section 5.1. Further analysis of the cases where $r = 1$ and $r > 1$ can be found in sections 5.2 and 5.3, respectively.

5.1 Cone of tangents and the normal cone

A valid linear 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 [25]). A more general convex set may not have any facets; for example, all the nontrivial faces of the unit ball in $\mathbb{R}^k$ have dimension 0. In order to gauge the strength of a supporting hyperplane, we propose looking at the face of the tangent cone given by its intersection with the supporting hyperplane. For a polyhedron, this gives the face of the polyhedron defined by the supporting hyperplane; for the unit ball in $\mathbb{R}^k$, the intersection of the (unique) supporting hyperplane at a boundary point x with the tangent cone at x has dimension $k-1$. We formalize the idea of a face of the tangent cone in the following definition.

Definition 1 *Let Y be a nonempty closed convex set in $I\!\!R^k$. For any point $\bar{y} \in I\!\!R^k$, 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 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*

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

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

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

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 $\{u : u = \check{y} + d, d \in \mathit{tcf}(\check{y}, Y, H)\}$ contains the face $H \cap Y$ of Y.

The equivalence of the two definitions of tcone follows from page 135 of Hiriart-Urruty and Lemaréchal [12]. The geometry of semidefinite programming is surveyed by Pataki [31]. Conceptually, the idea of a face of the tangent cone will be used as an analogue of the idea of a face of a polyhedron. In sections 5.2 and 5.3, we show that if Y is full-dimensional and if the inequality arises from an eigenvector of $\bar{S}$ with smallest eigenvalue then the dimension of the face of the tangent cone is related to the dimension of the corresponding eigenspace. We will illustrate these concepts with the following example:

Example 1 *Consider*

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

Consider the point $\bar{y} = [3 \quad 2]^T$ 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 $\hat{y} = [1 \quad 0]^T$. The point $\check{y} = (\bar{y} - \lambda \hat{y}) = [4 \ \ 2]^T \in Y$. This example is illustrated in figure 2. We have

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

The extreme rays of ncone$(\check{y}, Y)$ are $[-1 \ \ 0]$ and $[-5 \ \ 4]$, respectively. The first ray gives the linear constraint $y_1 \geq 4$ which gives the face of the tangent cone generated by d^1. The second

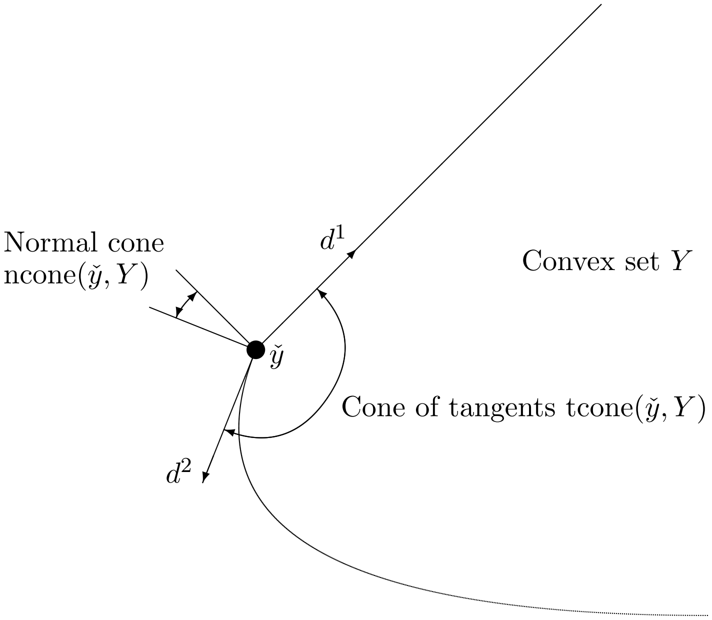
Normal cone
$\text{ncone}(\check{y}, Y)$
d^1
Convex set Y
$\check{y}$
Cone of tangents $\text{tcone}(\check{y}, Y)$
d^2

Figure 2: The tangent cone and normal cone for Example 1

ray gives the linear constraint $5y_1 - 4y_2 \geq 12$ and the face of the tangent cone defined by this constraint at $\breve{y}$ is the halfline generated by d^2. This second ray defines a face of dimension 0 of the feasible region Y, but a face of dimension $k-1$ of the tangent cone. It is desirable to add constraints corresponding to higher dimensional faces of tcone$(\breve{y}, Y)$, rather than weaker constraints that are active at $\breve{y}$, such as $2y_1 - y_2 \geq 6$. The latter constraint is obtained from the internal ray $[-2 \;\; 1]$ in ncone$(\breve{y}, Y)$.

In general convex programming, extreme rays of the normal cone may produce the most effective cuts, because of the duality between the normal cone and the tangent cone for convex sets [3, 35]. We characterize the normal cone for our problem and thus derive expressions for effective cuts; we will use these characterizations in Theorem 9 to obtain a lower bound on the dimension of the face of the tangent cone defined by a hyperplane. We will now derive an expression for ncone$(\breve{y}, Y)$.

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

$$\begin{array}{rcl} ncone(\breve{y}, Y) & = & cone(\partial\lambda_{\max}(\mathcal{A}^T\breve{y} - C)) \\ & = & \{\mathcal{A}(Q_2 V Q_2^T) : \;\; V \succeq 0\} \end{array} \qquad (12)$$

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

Proof: The proof for the first expression for ncone$(\breve{y}, Y)$ can be found in theorem 1.3.5 on page 245 of Hiriart-Urruty and Lemaréchal [12]. Using the second expression (5) for the subdifferential $\partial\lambda_{\max}(\mathcal{A}^T\breve{y} - C)$, we have

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

□

In general, the cone $\{\mathcal{A}(Q_2 V Q_2^T) : V \succeq 0\}$ may not be closed, as in the following example drawn from Pataki [32]:

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

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

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 [32]) to guarantee the closure of the cone $\{\mathcal{A}(Q_2 V Q_2^T) : V \succeq 0\}$ is the existence of a $y \in I\!R^k$ 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 tcone$(\breve{y}, Y)$.

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*

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

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

Proof: We have

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

□

We note that an alternative derivation for the tangent cone for the cone of positive semidefinite matrices can be found in Bonnans and Shapiro [4].

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 face of the tangent cone 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 η, then the constraint $\eta^T(C - \mathcal{A}^T y)\eta \geq 0$ defines a facet of the tangent cone of Y at $\check{y}$.*

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

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

where $Q = [Q_1 \ \eta]$ is an orthogonal matrix and Λ 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 $\eta^T(\check{S} - \mathcal{A}^T d)\eta \geq 0$, or equivalently as

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

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

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

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

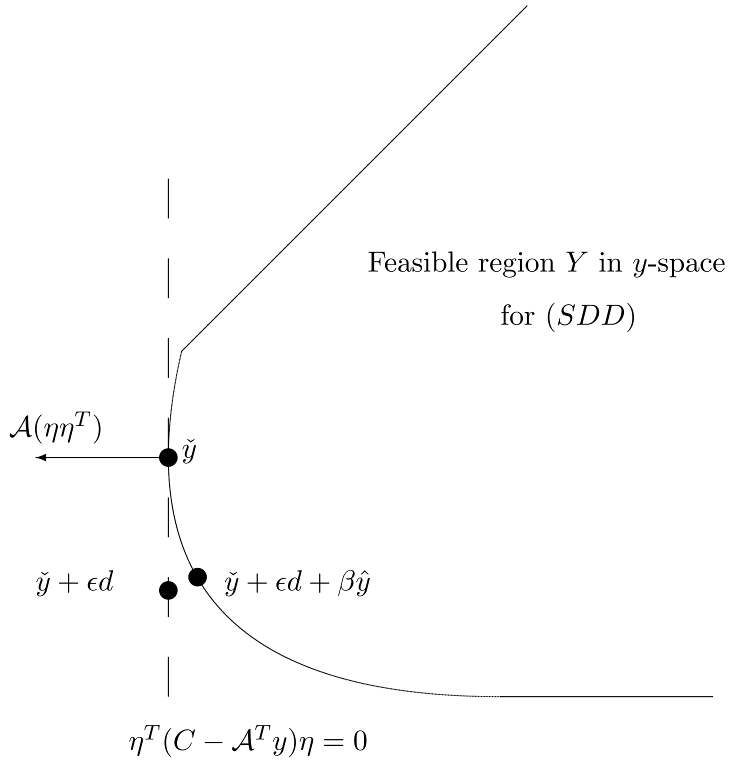
Feasible region Y in y-space
for (SDD)
$\mathcal{A}(\eta\eta^T)$
$\check{y}$
$\check{y}+\epsilon d$
$\check{y}+\epsilon d+\beta\hat{y}$
$\eta^T(C-\mathcal{A}^Ty)\eta=0$

Figure 3: The tangent plane defined by the cutting plane in y-space

When the multiplicity of the maximum eigenvalue is one, $\lambda_{\max}(A^Ty-C)$ 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 $\sum_{i=1}^k d_i\eta^TA_i\eta\leq 0$ whose boundary is the only supporting hyperplane at $y=\check{y}$ given by (15). For an illustration of this result, see figure 3.

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, and this nullity is typically greater than one. In exact arithmetic, the nullity of $\check{S}$ may be one as Algorithm 1 converges to an optimal solution to (SDD), but there are likely to be several very small eigenvalues. Computationally, the effective nullity of $\check{S}$ may be greater than one. 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 face of the tangent cone. Let $\check{S}$ have an eigendecomposition that is given by (11). We let r denote the nullity of $\check{S}$, and η denotes any vector of norm one in the nullspace of $\check{S}$. We now obtain a lower bound on the dimension of the face of the tangent cone.

Proposition 1 *The dimension of the face of the tangent cone to Y at $\check{y}$ defined by the valid* *constraint* $\eta^T(C - \mathcal{A}^T y)\eta \geq 0$ *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

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

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

In section 6, we give examples where the face of the tangent cone has dimension far greater than that implied by Proposition 1. Eigenvectors η that make the dimension of the corresponding tangent plane as large as possible give the best linear approximation to the cone of tangents. We now characterize the vectors η which will do this, leading to the strengthened result of Theorem 9. 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 use $\hat{y}$ to define one particular slice as

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

We first show that Π is a closed and bounded set and that every nonzero $v \in \text{ncone}(\check{y}, Y)$ can be scaled to give a point in Π.

Proposition 2 *For any nonzero* $v \in$ *ncone*$(\check{y}, Y)$*, the inner product* $\hat{y}^T v$ *is strictly positive. Further, the set* Π *is compact.*

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

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

Therefore,

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

We note that Π is the image under a linear transformation of the closed and bounded set $\{V \succeq 0 : \text{trace}(V) = 1\}$, so Π must also be closed and bounded. □

It follows from Proposition 2 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 8 *The extreme rays of $ncone(\check{y}, Y)$ 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

$$\begin{array}{llcrcl}
\min & & & g^T v & & \\
\text{subject to} & \mathcal{A}(Q_2 V Q_2^T) & - & v & = & 0 \qquad (SP8) \\
 & & & \hat{y}^T v & = & 1 \\
 & V & & & \succeq & 0
\end{array}$$

for some vector g. The dual of this problem is

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

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

$$\begin{array}{llcl}
\max & z & & \\
\text{subject to} & zI & \preceq & Q_2^T(\mathcal{A}^T g)Q_2. \qquad (SD8)
\end{array}$$

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 V must be in the nullspace of the optimal dual slack matrix, giving the result stated in the theorem. □

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

Theorem 9 *Let v be an extreme ray found by solving $(SP8)$, and assume the nullity of the optimal slack matrix in $(SD8)$ is one. The face of the tangent cone defined by the constraint $v^T y \leq v^T \check{y}$ has dimension at least $k - r$.*

Proof: Let u be a basis for the nullspace of the optimal slack matrix for $(SD8)$. From complementary slackness for the $(SP8)$ and $(SD8)$ 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 face of the tangent cone 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 $(SP8)$ and let $\tilde{d} = \hat{y}z - g$. From complementary slackness in the pair of semidefinite programs in theorem 8 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 face of the tangent cone defined by the constraint. The vector u is in the nullspace 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

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

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

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

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 $d' \in I\!R^k$ 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. □

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 [13]). 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

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

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

6 Examples of the face of the tangent cone

In this section, we look at examples of faces of the tangent cone for some semidefinite programs. The first example in section 6.1 considers a simple second order cone constraint of size 2. Our aim is to use this example to illustrate the power of Theorem 8 in generating stronger cutting planes. A more complicated non-polyhedral example with a large value of r is the subject of section 6.2.

6.1 SOCP example

The second order cone constraint $y_1^2 \geq y_2^2$, $y_1 \geq 0$, can be represented as the SDP constraint

$$y_1 \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + y_2 \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \succeq 0.$$

Consider the point $\bar{y} = [-1, 0]$. The dual slack matrix $S = \begin{bmatrix} y_1 & y_2 \\ y_2 & y_1 \end{bmatrix}$ has a negative eigenvalue -1 with multiplicity 2 at $\bar{y}$. Moreover, the eigenvector corresponding to this eigenvalue is not unique. Note that $\hat{y}$ from Lemma 1 is $[1, 0]^T$, and $\check{y}$ from (10) is $[0, 0]^T$. One choice for a cutting plane would be to choose any eigenvector corresponding to the minimum eigenvalue at $\bar{S}$. Applying the QR algorithm gives the following two choices for the eigenvector: (1) $\eta_1 = [1, 0]^T$ that gives the cutting plane $y_1 \geq 0$, and (2) $\eta_2 = [0, 1]^T$ that again gives the cutting plane $y_1 \geq 0$. Note that the face of the tangent cone defined by this constraint at this origin is the origin itself, so has dimension 0. Incidentally, these are the eigenvectors returned by the *eig* routine in our implementation of the cutting plane method in MATLAB.

The cone of tangents at the origin is

$$\text{tcone } = \{d \in I\!R^2 : d_1 \geq |d_2|\}$$

and the normal cone is

$$\text{ncone } = \{d \in I\!R^2 : d_1 \leq -|d_2|\}.$$

At the origin, we have $Q_2 = I$. Given a vector $u \in I\!R^2$ with $||u||_2^2 = 1$, we obtain

$$\mathcal{A}(Q_2 u u^T Q_2^T) = \begin{bmatrix} -1 \\ -2u_1 u_2 \end{bmatrix}$$

and the extreme rays of the normal cone are obtained by taking $u_1 = \pm\frac{1}{\sqrt{2}}$ and $u_2 = \pm\frac{1}{\sqrt{2}}$.

By Theorem 8, we can obtain the extreme rays by first picking a vector $g \in I\!R^2$ and then finding an eigenvector of minimum eigenvalue of the matrix

$$\mathcal{A}^T g := g_1 \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix} + g_2 \begin{bmatrix} 0 & -1 \\ -1 & 0 \end{bmatrix} = \begin{bmatrix} -g_1 & -g_2 \\ -g_2 & -g_1 \end{bmatrix}.$$

The eigenvalues of this matrix are $-g_1 \pm g_2$. We have two cases:

$g_2 > 0$**:** The minimum eigenvalue is $-g_1 - g_2$ and the corresponding eigenvector is $u = [1,\ 1]^T$, normalized. Then

$$\mathcal{A}(Q_2 u u^T Q_2^T) = \begin{bmatrix} -1 \\ -1 \end{bmatrix},$$

one of the extreme rays of the normal cone. This gives the cutting plane $y_1 + y_2 \geq 0$. The face of the tangent cone defined by this constraint at the origin is the ray $y_1 + y_2 = 0$ that has dimension 1.

$g_2 < 0$**:** The minimum eigenvalue is $-g_1 + g_2$ and the corresponding eigenvector is $u = [1,\ -1]^T$, normalized. Then

$$\mathcal{A}(Q_2 u u^T Q_2^T) = \begin{bmatrix} -1 \\ 1 \end{bmatrix},$$

the other extreme ray of the normal cone. This gives the cutting plane $y_1 - y_2 \geq 0$. The face of the tangent cone defined by this constraint at the origin is the ray $y_1 = y_2$ that has dimension 1.

Hence, *almost every choice of g leads to an extreme ray of the normal cone.* The exception is to take $g_2 = 0$. This returns the weaker cutting plane $y_1 \geq 0$, which as we discussed before, defines a face of the tangent cone of dimension 0.

The result of Theorem 9 may underestimate the dimension of the face of the tangent cone defined by the constraint. For this example, we have $k - r = 0$, and the dimension of the face of the tangent cone is 1.

6.2 Non-polyhedral 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 $\eta = e_i - e_j$ 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 face of the tangent cone 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 (SP8) 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 face of the tangent cone 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 non-polyhedral nature of $\text{tcone}(\check{y}, Y)$. The choice of $g = [1 \ \ 1 \ \ 0]$ in (SP8) 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 face of the tangent cone of dimension 0. Similarly, for $n \geq 3$ every hyperplane $d_i + d_j = 0$, $i \neq j$ defines a face of the tangent cone 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 face of the tangent cone 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 face of the tangent cone $\{d \in \mathbb{R}^4 : \ d_1 = 0, \ d_2 = 0, \ d_3 = 0, \ d_4 \geq 0\}$, 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 face of the tangent cone of dimension $n - 3$.

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

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

Since $n \geq 4$, at least two components of η 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 face of the tangent cone of dimension 0 which is strictly smaller than $k - r$.

7 Computational Results

In this section, we compare the effect of adding a cutting plane corresponding to the minimum eigenvalue of $\bar{S} = (C - \mathcal{A}^T \bar{y})$ with adding other cutting planes. We test the approach on Ising spin glass MaxCut problems with 10000 vertices, taken from [20]. We use CPLEX 12 to solve the LP relaxations on one core of an Apple Mac Pro with a 2x2.8 GHz Quad-Core Intel Xeon processor. The most negative eigenvalue was found using the power method, applied first to find the largest eigenvalue and then applied to a shifted version of $\bar{S}$ to find the most negative eigenvalue and a corresponding eigenvector. The tested alternative to adding this eigenvector was to sparsify it, ensuring that it still gave a violated constraint. This is a similar approach to that in [33]. We investigated adding either 10 or 20 sparse constraints at each iteration.

In Table 1, we compare the average improvement in the lower bound on the SDP relaxation for three instances, after 20 iterations and after 100 iterations. The initial lower bound is obtained using two sets of dual constraints: (i) the diagonal entries of S are nonnegative, and (ii) for each 2×2 principal minor M of S with nonzero off-diagonal, we impose the constraint that $d^T M d \geq 0$, where $d = (1, -1)^T$. For our test problems, the optimal solution to this relaxation results in each M being positive semidefinite.

It is clear that the single eigenvector is giving better bounds than 10 or 20 sparse constraints. The linear programs with the sparse constraints solve more quickly than the LPs with dense constraints, so we also compare the two approaches when equalizing for the time used by CPLEX. In this comparison, the sparse approaches are better. However, this comparison ignores the time required to find the sparse constraints, which is far greater than the CPLEX time for these instances; in terms of total time, the dense eigenvector approach is far faster. After 500 iterations, the approach of adding just the constraint coming directly from the most negative eigenvalue improves the lower bound by approximately 78%, while using total time comparable to 100 iterations of the methods adding 10 or 20 sparse constraints at a time.

The results in Section 5 apply in the case when the multiplicity of the minimum eigenvalue of the dual slack matrix is greater than one. This is difficult to simulate in an actual computational experiment which is why we have not incorporated it in the computational results. An upper bound for the multiplicity for the most negative eigenvalue of the dual

	Eigenvector	10 sparse constraints	20 sparse constraints
20 iterations	6.1	2.7	3.0
100 iterations	26.9	11.5	12.4
10 secs CPLEX improv	4.7	14.2	14.2
10 secs CPLEX iters	17	133	117

Table 1: Percentage improvement in lower bound when adding a constraint corresponding to the most negative eigenvalue or adding constraints corresponding to sparse versions of this eigenvector. The last row notes the number of iterations performed while using a total of 10 seconds of CPU time solving LPs.

slack matrix is given by Theorem 2. This multiplicity tends to increase towards optimality since the smaller negative eigenvalues tend to coalesce together. This is why we believe that the results in Section 5 are important from a practical point of view as well.

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. Further computational results with the interior point cutting plane algorithm described in this paper can be found in Sivaramakrishnan et al. [38].

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. We proposed gauging the strength of a cutting plane by examining the dimension of the face of the tangent cone induced by the cut. 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. This cutting plane is constructed from the unique eigenvector corresponding to the most negative eigenvalue of the dual slack matrix. The computational results in section 7 show that this cutting plane is stronger than other valid cutting planes that are generated from sparse versions of the eigenvector. Further, for higher values of the nullity, cutting planes that induce a higher dimensional face of the tangent cone can be found by determining the smallest eigenvalue and corresponding eigenvector of the matrix given in Theorem 8. 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.

Finally, the results in section 5.3 can also be used in other cutting plane algorithms such as the spectral bundle method of Helmberg [11, 10], and the non-polyhedral cutting surface algorithms of Sivaramakrishnan et al. [38] and Oskoorouchi et al. [27, 28]; when the most negative eigenvalue of the dual slack matrix during the course of the algorithm has multiplicity greater than 1. In this case, one can generate vectors that are extreme rays of the normal cone in order to update the cutting plane model by solving the simple eigenvalue problem in Theorem 8. This will improve the convergence of these algorithms, since these vectors provide stronger cutting planes than a naive black-box that simply computes an eigenvector corresponding to the minimum eigenvalue of the dual slack matrix.

Acknowledgements

We are grateful to a referee for carefully reading the paper and for comments and suggestions that have improved the presentation of the results in the paper.

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. Towards a practical volumetric cutting plane method for convex programming. *SIAM Journal on Optimization*, 9(1):190–206, 1999.

[3] D. P. Bertsekas, A. Nedic, and A. E. Ozdaglar. *Convex Analysis and Optimization.* Athena Scientific, 2003.

[4] J. F. Bonnans and A. Shapiro. *Perturbation Analysis of Optimization Problems.* Springer-Verlag, New York, 2001.

[5] 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.

[6] 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.

[7] E. de Klerk. *Aspects of Semidefinite Programming: Interior Point Algorithms and Selected Applications*, volume 65 of *Applied Optimization.* Kluwer Academic Publishers, Dordrecht, Netherlands, 2002.

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

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

[10] C. Helmberg. Numerical evaluation of SBmethod. *Mathematical Programming*, 95(2):381–406, 2003.

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

[12] J.-B. Hiriart-Urruty and C. Lemaréchal. *Convex Analysis and Minimization Algorithms I.* Springer-Verlag, Berlin, 1993.

[13] J.-B. Hiriart-Urruty and C. Lemaréchal. *Convex Analysis and Minimization Algorithms II.* 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. 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.

[17] 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.

[18] 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.

[19] 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.

[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. *Optimization Methods and Software*, 18(5):507–534, 2003.

[22] J. E. Mitchell and M. J. Todd. Solving combinatorial optimization problems using Karmarkar's algorithm. *Mathematical Programming*, 56(3):245–284, 1992.

[23] R. D. C. Monteiro. First- and second-order methods for semidefinite programming. *Mathematical Programming*, 97(1–2):209–244, 2003.

[24] M. V. Nayakkankuppam. Solving large-scale semidefinite programs in parallel. *Mathematical Programming*, 109(2-3):477–504, 2007.

[25] G. L. Nemhauser and L. A. Wolsey. *Integer and Combinatorial Optimization.* John Wiley, New York, 1988.

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

[27] M. R. Oskoorouchi and J.-L. Goffin. A matrix generation approach for eigenvalue optimization. *Mathematical Programming*, 109(1):155–179, 2007.

[28] M. R. Oskoorouchi and J. E. Mitchell. A second-order cone cutting surface method: complexity and application. *Computational Optimization and Applications*, 43(3):379–409, 2009.

[29] M. Overton. Large-scale optimization of eigenvalues. *SIAM Journal on Optimization*, 2(1):88–120, 1992.

[30] 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.

[31] 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.

[32] G. Pataki. On the closedness of the linear image of a closed convex cone. *Mathematics of Operations Research*, 32(2):395–412, 2007.

[33] A. Qualizza, P. Belotti, and F. Margot. Linear programming relaxations of quadratically constrained quadratic programs. In J. Lee and S. Leyffer, editors, *IMA Volumes in Mathematics and its Applications*, volume 154, pages 407–426. Springer, 2012.

[34] 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.

[35] R. T. Rockafellar. *Convex Analysis.* Princeton University Press, Princeton, NJ, 1970.

[36] 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.

[37] K. K. Sivaramakrishnan. A parallel interior point decomposition algorithm for block angular semidefinite programs. *Computational Optimization and Applications*, 46(1):1–29, 2010.

[38] K. K. 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.

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

[40] K. C. Toh, M. J. Todd, and R. Tutuncu. On the implementation and usage of SDPT3 – a Matlab software package for semidefinite-quadratic-linear programming, version 4.0. Technical report, Department of Mathematics, National University of Singapore, Singapore, June 2010.

[41] P. M. Vaidya. A new algorithm for minimizing convex functions over convex sets. *Mathematical Programming*, 73(3):291–341, 1996.

[42] M. Yamashita, K. Fujisawa, and M. Kojima. SDPARA: Semidefinite programming algorithm parallel version. *Parallel Computing*, 29(8):1053–1067, 2003.

[43] 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.