A unifying framework for several cutting plane methods for semidefinite programming

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

Joint work with **Kartik Krishnan**.

18 February 2004

Abstract

Cutting plane methods provide the means to solve large scale semidefinite programs (SDP) cheaply and quickly. We give a survey of various cutting plane approaches for SDP in this paper. These cutting plane approaches arise from various perspectives, and include techniques based on interior point cutting plane approaches, and an approach which mimics the simplex method for linear programming. We present an accessible introduction to various cutting plane approaches that have appeared in the literature. We place these methods in a unifying framework which illustrates how each approach arises as a natural enhancement of a primordial linear programming cutting plane scheme based on a semi-infinite formulation of the SDP.

Contents

1 Semidefinite programming

Standard form:

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

with dual

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

- C, X and S are square symmetric matrices.
- Require X and S be **positive semidefinite** (psd).
- Inner products are the **Frobenius inner product**, so $C \bullet X = \text{trace}(CX)$.
- $\mathcal{A}X = \begin{bmatrix} A_1 \bullet X \\ \vdots \\ A_k \bullet X \end{bmatrix}$, $\mathcal{A}^T y = \Sigma_{i=1}^{k} y_j A_j$, where each A_j is a symmetric matrix.
- Denote

$$Y := \{y \in I\!R^m : C - \mathcal{A}^T y \succeq 0\}.$$

Applications

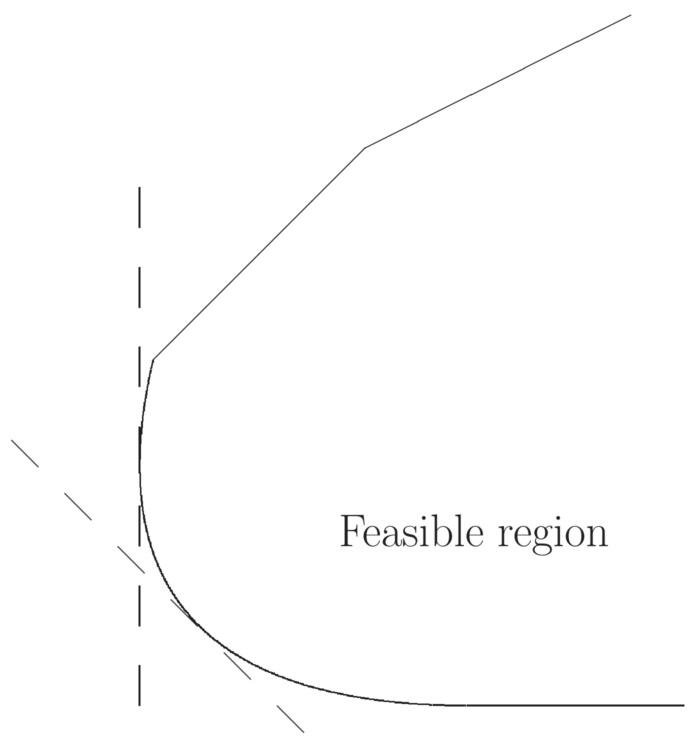
- control
- developing approximation algorithms for combinatorial optimization problems
- finance
- statistics
- ...

Methods for solving SDPs

- **Primal-dual interior point approaches** are limited in the size of problems they can solve.
- Alternatives include:
  - The **bundle method** of Helmberg, Rendl, and Kiwiel.
  - The **nonlinear programming approach** of Burer, Monteiro, and Zhang.
  - Methods based on **cutting plane models**.

2 A semi-infinite linear program

- (SDP) and (SDD) are **convex programming** problems.
- Only nonlinearity is the positive semidefiniteness (PSD) requirement.
- **Replace PSD constraint by linear constraints.**



Dual formulation:

$$\begin{array}{lrcl} \max & & b^T y & & (LDD) \\ \text{subject to} & d^T(C - \mathcal{A}^T y)d & \geq & 0 \quad \forall \text{d with } ||d||^2 = 1. \end{array}$$

- If (SDP) has m constraints, then (LDD) has m variables.
- Need to be **selective** with the vectors d included in the constraints. For example, use a **cutting plane approach** to select vectors d.
- Typically, we will need some vectors d that are **dense**, leading to a dense linear programming problem.
- Have far fewer variables than in a primal formulation, so get smaller linear programs (although they are still dense).

Regaining a primal solution

- We only use a finite number of vectors d, so we are solving a **relaxation** of the dual:

$$\begin{array}{lrcl} \max & b^T y & & (LDR) \\ \text{subject to} & d_i^T (C - \mathcal{A}^T y) d_i & \geq & 0 \\ & & & \text{for vectors } d_i, i = 1, \ldots, m. \end{array}$$

- The optimal value to (LDR) gives an upper bound on the optimal value of (SDP).

- (LDR) is a relaxation of (SDD). The dual linear program to (LDR) is a **constrained version** of (SDP).

- (LDR) can be rewritten as

$$\begin{array}{lrcl} \max & b^T y & & \\ \text{subject to} & \Sigma_{j=1}^k y_j (d_i^T A_j d_i) & \leq & d_i^T C d_i \\ & & & \text{for vectors } d_i, i = 1, \ldots, m. \end{array} \qquad (LDR)$$

- Writing the linear programming dual to (LDR) directly gives

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

- Now,

$$d_i^T C d_i = \text{trace}(d_i^T C d_i) = \text{trace}(C d_i d_i^T) = C \bullet d_i d_i^T.$$

- So (LPR) can be rewritten as

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

- Thus, any feasible solution to (LPR) gives a **feasible solution to (SDP)**, $X = \Sigma_{i=1}^{m} x_i d_i d_i^T$

- Note that

$$
\begin{array}{rcl}
X & = & \Sigma_{i=1}^m x_i d_i d_i^T \\
& = & \begin{bmatrix} | & & | \\ d_1 & \ldots & d_m \\ | & & | \end{bmatrix} \begin{bmatrix} x_1 & & \\ & \ddots & \\ & & x_m \end{bmatrix} \begin{bmatrix} - & d_1^T & - \\ & \vdots & \\ - & d_m & - \end{bmatrix} \\
& =: & \qquad D \qquad\qquad\qquad M \qquad\qquad\qquad D^T
\end{array}
$$

- Thus (LPR) can be rewritten as

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

A perfect set of constraints

Let X^* solve (SDP) and y^*, S^* solve (SDD).

- X^* has an **eigendecomposition**:

$$X^* = [P \;\; Q] \begin{bmatrix} \Lambda & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} P^T \\ Q^T \end{bmatrix}$$

- X^* and S^* are **simultaneously diagonalizable**:

$$S^* = [P \;\; Q] \begin{bmatrix} 0 & 0 \\ 0 & \Gamma \end{bmatrix} \begin{bmatrix} P^T \\ Q^T \end{bmatrix}$$

- Let $p^1, \ldots, p^r$ be the columns of P. Then the optimal solutions to

$$\begin{array}{lrcl} \max & b^T y & & \\ \text{subject to} & {p^i}^T (C - \mathcal{A}^T y) p^i & \geq & 0, \;\; i = 1, \ldots, r \end{array}$$

  and its dual give the **optimal solutions** to (SDD) and (SDP): set $x = \lambda$, where $\lambda = \text{diag}(\Lambda)$.

- Pataki: expect $r \approx O(\sqrt{k})$.

3 Cutting plane algorithm for (SDP)

- Choose an **initial set** of constraints for (LDR).
- **Solve** (LDR) and (LPR) approximately using an interior point method.
- Get **trial point** $\bar{y}$.
- If $C - \mathcal{A}^T\bar{y}$ is not psd, find vectors d and corresponding **violated constraints** $d^T(C - \mathcal{A}^T y)d \geq 0$.
- These vectors d can be **eigenvectors** of $C - \mathcal{A}^T\bar{y}$ with negative eigenvalue.
- Modify (LDR) and (LPR), and **repeat**.

4 Nonpolyhedral cutting plane models

Recall that (LPR) can be rewritten as

$$\begin{array}{lrcll} \min & C \bullet (DMD^T) & & & \\ \text{s.t.} & A_j \bullet (DMD^T) & = & b_j & j = 1, \ldots, m \\ & M & \succeq & 0 & \\ & M & \multicolumn{3}{l}{\textbf{diagonal}} \end{array} \tag{1}$$

Here $M \in \mathcal{S}^m$, and $D \in I\!R^{n \times m}$ with jth column d_j.
If the columns of D contain **eigenbases** for all the strictly positive eigenvalues of X^*, then the solution to (1) is an exact solution to (SDP).
In other cases, a solution provides an upper bound on this objective value.

Drop requirement that M be diagonal

Consider a relaxation of (1) dropping the requirement that M be diagonal.

$$\begin{array}{lrcl} \min & C \bullet (DMD^T) & & \\ \text{s.t.} & A_j \bullet (DMD^T) & = & b_j \quad j = 1, \ldots, m \\ & M & \succeq & 0 \end{array} \tag{2}$$

with $M \in \mathcal{S}^m$, and $D \in \mathbb{R}^{n \times m}$.

If the rank of D is n, then (2) is essentially (SDP).

In fact, if Range(D) $\supset$ Range(X^*), then a solution to (2) is an exact solution to (SDP).

This is a **less stringent** requirement than the polyhedral cutting plane model, where we require the exact eigenvectors of X^*.

But this is a semidefinite program, so harder to solve than the linear program (LPR).

Necessary to **limit the number of columns** in D to keep the algorithm practical.

The dual problem

The dual to (2) is

$$\begin{array}{lcl} \max & b^T y & \\ \text{subject to} & D^T(C - \mathcal{A}^T y)D & \succeq 0 \end{array} \qquad (3)$$

If D has rank n then this is equivalent to the original problem (SDD).

The linear programming formulation (LDR) requires only that the **diagonal entries** of $D^T(C - \mathcal{A}^T y)D$ be non-negative.

A block-diagonal model

Finally, another relaxation of (1) which is more restrictive than (2) is the following.

$$\begin{array}{lrcll} \min & C \bullet (DMD^T) & & & \\ \text{s.t.} & A_j \bullet (DMD^T) & = & b_j & j = 1, \ldots, m \\ & M & \succeq & 0 & \\ & M & \multicolumn{2}{l}{\textbf{block diagonal}} & \end{array} \qquad (4)$$

Dual:

$$\begin{array}{lrcl} \max & & b^T y & \\ \text{subject to} & D_i^T (C - \mathcal{A}^T y) D_i & \succeq & 0 \end{array} \qquad (5)$$

where each D_i corresponds to a diagonal block of M.

5 Approximately solving the dual subproblem

- Need to find good cutting planes.
- Useful to search for cutting planes before solving the current dual problem to optimality.
- This should lead to deeper cuts, with more of the feasible region removed.

Generic representation of the dual

Write the dual relaxation as

$$\begin{array}{lrcl} \max & b^T y & & \\ \text{s.t.} & D_i^T(C - \mathcal{A}^T y)D_i & \succeq & 0 \quad i \in \mathcal{D} \end{array} \tag{6}$$

where $\mathcal{D}$ represents our current collection of constraints.

- If we add linear constraints, each D_i is a column vector.
- If we use the block diagonal formulation, each D_i is a matrix with a small number of columns.
- If we drop the restrictions on M, $\mathcal{D}$ consists of just one constraint. Then D_i is a matrix with approximately $\sqrt{k}$ columns, typically.

Methods for approximate solution

- Solve the relaxation **approximately using an interior point method**.
- Use a **bundle method**:
  have a proximity term $||y - \hat{y}||^2$ in the objective to keep iterates close to the old iterate $\hat{y}$.

Generic algorithm

1. Choose an initial finite set $\mathcal{D} = \{D_i\}$.
2. Approximately solve the current dual relaxation to get $\bar{y}$.
3. If $\bar{S} = (C - \mathcal{A}^T\bar{y}) \succeq 0$:
   - If duality gap small enough, STOP.
   - Else, solve the current dual relaxation more accurately.
4. Else find $D \in I\!R^{n\times r}$, with $r \leq \sqrt{2k}$ such that $D^T(C - \mathcal{A}^T\bar{y})D \not\succeq 0$.
5. Either add D to $\mathcal{D}$, or aggregate D into $\mathcal{D}$ and return to Step 2.

Algorithm	Model	u	Form of $\mathcal{D}$	Add or aggregate?
1	Polyhedral	zero	$D \in \mathbb{R}^{n\times 1}$ only	add, letting $\lvert\mathcal{D}\rvert$ grow
2	Polyhedral bundle	positive	$D \in \mathbb{R}^{n\times 1}$ only	add, letting $\lvert\mathcal{D}\rvert$ grow
3	Nonlinear Block-Diag	zero	$D \in \mathbb{R}^{n\times r}$ with $r \leq \sqrt{2m}$	add, letting $\lvert\mathcal{D}\rvert$ grow
4	Spectral bundle	positive	$D \in \mathbb{R}^{n\times r}$	aggregate, keeping $\lvert\mathcal{D}\rvert = 1$
5	Primal active set	zero	$D \in \mathbb{R}^{n\times r}$	aggregate, keeping $\lvert\mathcal{D}\rvert = 1$

6 Properties and a dual cone

Assume:

1. The constraints $\mathcal{A}(X) = b$ imply $\textbf{trace}(X) = a$ for some fixed a.

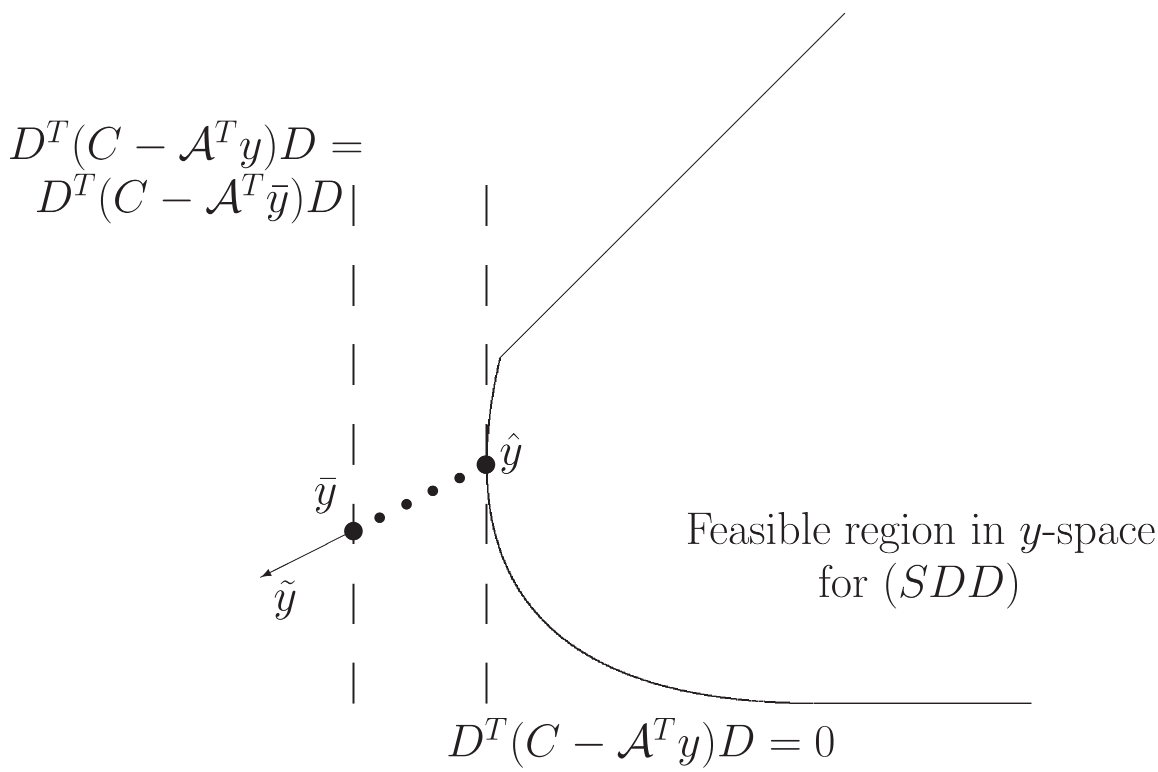
2. Given $\bar{y}$ and $\bar{S} = C - \mathcal{A}^T\bar{y}$, the column(s) of the cutting matrix D are eigenvectors of $\bar{S}$ corresponding to the **most negative eigenvalue** $\lambda_{\min}$ of $\bar{S}$.

Then:

There exists $\hat{y}$ feasible in (SDD) satisfying the constraint $D^T(C - \mathcal{A}^T y)D = 0$.

Constraint satisfied at equality:

- Since $\text{trace}(X) = a$, there exists $\tilde{y}$ with $\mathcal{A}^T\tilde{y} = I$.
- Let $\hat{y} := \bar{y} + \lambda_{\min}\tilde{y}$.
- Then $\hat{S} := C - \mathcal{A}^T\hat{y}$ is psd, and $D^T\hat{S}D = 0$.



Tangent space and cone of tangents

Take $Y \subseteq \mathbb{R}^2$ to be the vectors y that make the following matrix S positive semidefinite:

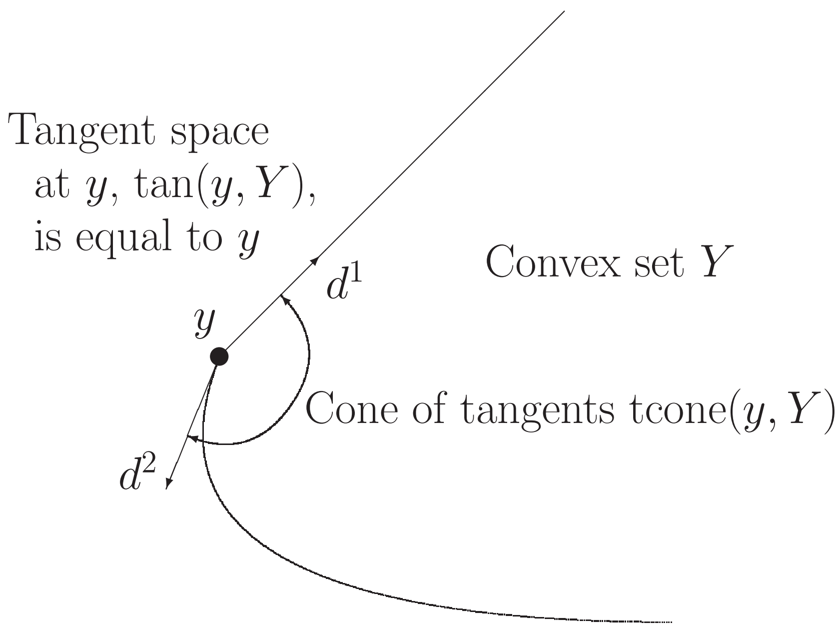
$$S = \begin{bmatrix} y_1 & y_2 & 0 \\ y_2 & y_1 - 3 & 0 \\ 0 & 0 & 2 - y_2 \end{bmatrix}.$$

Let $\bar{y} = [4 \quad 2]^T \in Y$. The corresponding matrix $\bar{S}$ has rank one and nullity two.

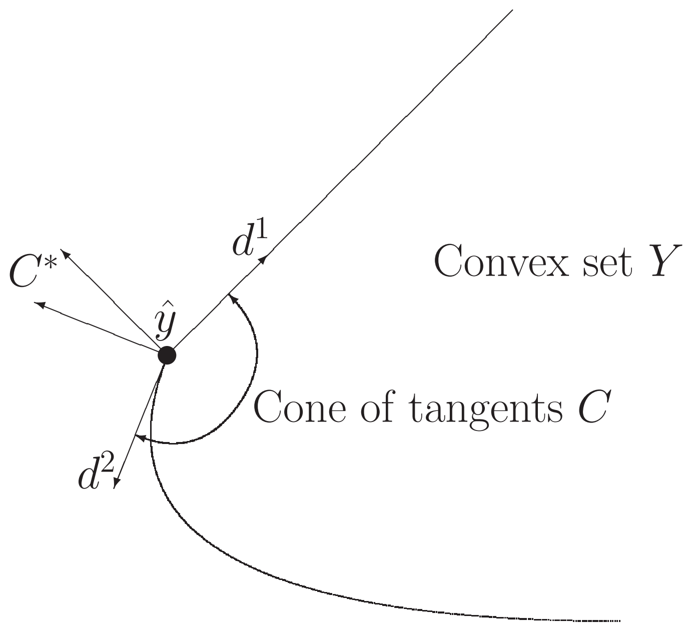
Basis for nullspace of $\bar{S}$: $[1, 2, 0]^T$, $[0, 0, 1]^T$.

These vectors give the constraints

$$\begin{aligned} y_2 &\leq 2 \\ 5y_1 - 4y_2 &\geq 12 \end{aligned}$$


Tangent space
at y, $\tan(y, Y)$,
is equal to y
d^1
Convex set Y
y
Cone of tangents $\mathrm{tcone}(y, Y)$
d^2

Dual cone



$$d_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \qquad d_2 = \begin{bmatrix} -4 \\ -5 \end{bmatrix}$$

Extreme rays of dual cone C^* give the constraints

$$\begin{aligned} y_2 &\leq 2 \\ 5y_1 - 4y_2 &\geq 12 \end{aligned}$$

More on the dual cone

- Other vectors in the nullspace of $\bar{S}$ will not give such strong constraints.
- In higher dimensions, the **dual cone may not be polyhedral**, and the number of extreme rays may be uncountable.
- If the columns of the matrix D form an eigenbasis for the minimum eigenvalue, then the constraint $D^T(C - \mathcal{A}^T y)D \succeq 0$ **represents exactly the cone of tangents** at $\hat{y}$.

Using an eigendecomposition

- Define **eigendecomposition** of $\hat{S}$:

$$\hat{S} = [P \;\; Q] \begin{bmatrix} 0 & 0 \\ 0 & \Gamma \end{bmatrix} \begin{bmatrix} P^T \\ Q^T \end{bmatrix}$$

- Note: we do not assume strict complementarity.

- (Alizadeh et al.)

  Tangent space of cone of psd matrices at $\hat{S}$:

  $$\{[P \;\; Q] \begin{bmatrix} 0 & V \\ V^T & U \end{bmatrix} \begin{bmatrix} P^T \\ Q^T \end{bmatrix} : U \in \mathcal{S}^{m-r}, V \in I\!R^{r \times (m-r)}\}$$

  After a step of length $O(\epsilon)$ in the tangent space, still within $O(\epsilon^2)$ of the cone of psd matrices.

The form of the of the dual cone

- In general, the dual cone C^* is equal to the closure of the set of vectors of the form

$$\tilde{C}^* := \{\mathcal{A}(PZP^T) : Z \succeq 0\}.$$

- The set $\tilde{C}^*$ may not be closed.

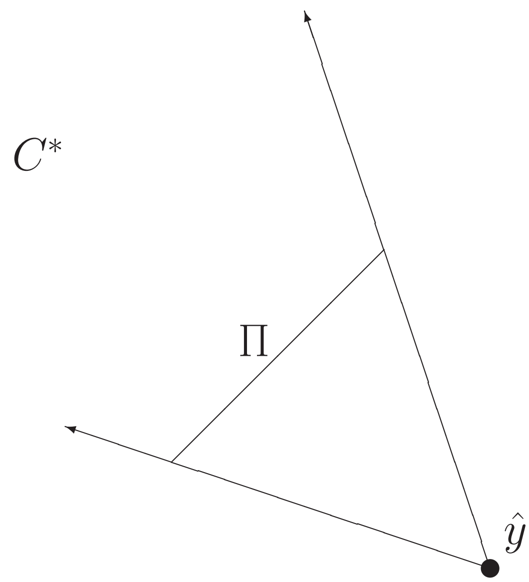
  Example (Pataki):

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

  The ray $(0,1)^T$ is not in the cone $\{\mathcal{A}(PZP^T) : Z \succeq 0\}$, but it is in its closure.

- Under our first assumption, the set $\tilde{C}^*$ is closed.

Finding the extreme rays of the dual cone



$$\Pi = \{\mathcal{A}(PZP^T) : \operatorname{trace}(Z) = 1, Z \succeq 0\}$$

- The extreme rays of C^* correspond to the extreme points of the compact set Π.
- Find these by minimizing an objective function over Π.

7 Complexity of the polyhedral cutting plane algorithm

- Theoretically, could use a **volumetric barrier cutting plane algorithm** to solve (SDD).
- Want to get within ϵ of optimality.
- Requires $O(m \log(\frac{1}{\epsilon}))$ **calls to the quadratic subproblem** and a similar number of Newton steps.
- Each **solution of the subproblem** requires $O(n^3)$ work if the QR algorithm is used to find the solutions to the subproblems. It also requires $O(mn^2)$ work to calculate $\bar{S}$.
- Each **Newton step** in (LDR) requires $O(m^3)$ work.
- Thus, for $m \geq n$, **total number of arithmetic operations** is of the order of $O(m^4 \log(\frac{1}{\epsilon}))$.
- A **primal dual SDP method** requires $O(mn^3 + m^2n^2 + m^3)$ arithmetic operations at each iteration and $O(\sqrt{n} \log(\frac{1}{\epsilon}))$ iterations. So overall complexity is $O(m^2n^{2.5} \log(\frac{1}{\epsilon}))$ for $n \leq m \leq n^2$. (Complexity can be reduced slightly if constraint matrices have special structure.)

8 Conclusions

1. Cutting plane approaches to the solution of semidefinite programs have attractive theoretical complexity.

2. High-dimensional cutting planes can be found efficiently.

3. The cone of tangents can be captured exactly through the use of a nonpolyhedral cutting plane.

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] 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.

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

[4] C. Helmberg and K. Kiwiel. A spectral bundle method with bounds. *Mathematical Programming*, 93(2):173–194, 2002.

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

[6] K. Krishnan and J. E. Mitchell. Properties of a cutting plane method for semidefinite programming. Technical report, Mathematical Sciences, Rensselaer Polytechnic Institute, Troy, NY 12180, May 2003.

[7] 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.