Towards a simplex-like method for conic programming

Kartik Krishnan

Department of Computing and Software
McMaster University

Joint work with Gábor Pataki (UNC) and Yin Zhang (Rice).
Thanks to Tamás Terlaky (Mac) and John Mitchell (RPI).

Math Colloquium, RPI, April 13, 2004

Overview

- – Conic optimization
  - Semidefinite optimization
  - Second Order Cone optimization
- Contrast active set and IPMs for conic optimization
- Primal simplex method for LP
- Notions of bfs and nondegeneracy in SDP
- Our simplex-like approach for SDP
- Preliminary computational results
- Conclusions and future work

Conic optimization

- **Primal**

$$\begin{aligned} \min\, & c^T x \\ Ax &= b \\ x &\in \mathcal{K} \qquad \text{convex, closed, non-empty interior, pointed} \end{aligned}$$

- **Dual**

$$\begin{aligned} \max\, & b^T y \\ A^T y + s &= c \\ s &\in \mathcal{K}^* \qquad \text{dual cone} \end{aligned}$$

- **Facts:**

  (1) Encompasses every convex optimization problem.

  (2) Variety of applications in science and engineering.

Special cones

- $\mathbb{R}^n_\oplus = \{x \in \mathbb{R}^n : x \geq 0\}$
  - Linear, self-dual
- $\mathcal{K}^q = \left\{x \in \mathbb{R}^n : x_1^2 \geq \|x_{2:n}\|^2, x_1 \geq 0\right\}$
  - Second order, self-dual
- $\mathbb{S}^n_\oplus = \{X \in \mathbb{S}^n : X \succeq 0\}$
  - Semidefinite, self-dual

Applications of conic programming

- Combinatorial optimization:
  Alizadeh, Goemans-Williamson, Lovasz-Schrijver, Ye.
- Linear control theory: Boyd, Vandenberghe, Balakrishnan, El Ghaoui, Feron.
- Signal Processing: Luo.
- Robust optimization: Ben Tal-Nemirovski, El Ghaoui.
- Copositive and polynomial programming: Lasserre, Parrilo, De Klerk, Pasechnik, Nesterov.
- Others include finance, statistics, moment problems etc.

SDP I.

- **Primal**

$$\begin{array}{rcl} \text{min} & C \bullet X & \\ A_i \bullet X & = & b_i,\ i = 1, \ldots, m, \qquad (SDP) \\ X & \succeq & 0, \end{array}$$

where A_i, C, $X \in \mathbb{S}^n$, $b \in \mathbb{R}^m$, $A \bullet B = \sum_{i=1}^n \sum_{j=1}^n A_{ij} B_{ij}$.

- **Dual**

$$\begin{array}{rcl} \text{max} & b^T y & \\ \sum_{i=1}^m y_i A_i + S & = & C, \qquad (SDD) \\ S & \succeq & 0, \end{array}$$

where $S \in \mathbb{S}^n$, $y \in \mathbb{R}^m$.

- **Assumptions:**
(1) Both (SDP) and (SDD) have strictly feasible points.
(2) A_i, $i = 1, \ldots, m$ are linearly independent in $\mathbb{S}^n$.
(3) $m = O(n)$.

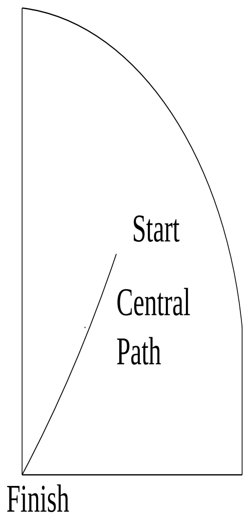
- **Alternative notation:** $\mathcal{A}X = (A_i \bullet X)_{i=1}^m$, $\mathcal{A}^* y = \sum_{i=1}^m y_i A_i$

SDP II.

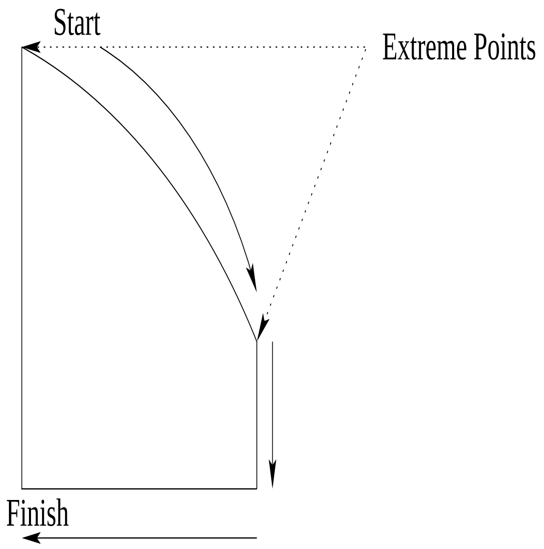
- **Optimality Conditions:**
  - Primal feasibility.
  - Dual feasibility.
  - Complementary slackness $XS = 0$.
- **Simultaneously Diagonalizable:** If X and S are optimal for (SDP) and (SDD) then $\exists$ orthogonal matrices $P_1 \in \mathbb{R}^{n\times r}$, $P_2 \in \mathbb{R}^{n\times(n-r)}$ and diagonal matrices $\Lambda \in \mathbb{S}^r_\oplus$, $\Omega \in \mathbb{S}^{(n-r)}_\oplus$ such that

$$\begin{aligned} X &= \begin{bmatrix} P_1 & P_2 \end{bmatrix} \begin{bmatrix} \Lambda & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} P_1^T \\ P_2^T \end{bmatrix} \\ S &= \begin{bmatrix} P_1 & P_2 \end{bmatrix} \begin{bmatrix} 0 & 0 \\ 0 & \Omega \end{bmatrix} \begin{bmatrix} P_1^T \\ P_2^T \end{bmatrix} \end{aligned}$$

Contrast simplex and IPMs for CP I.



Interior Point Method



Simplex Method

Interior Point Methods deal with matrices of full rank.

In the simplex method, the rank of the extreme points satisfy

(1) $r <= m$ (Linear Programming)

(2) $r(r+1)/2 <= m$ (Semidefinite Programming)

Contrast simplex and IPMs for CP II:

Why a simplex method for conic programming?.

1. Listed as an open problem in Alizadeh & Goldfarb (03) and Vandenberghe & Boyd (96).
2. Can do each iteration more quickly. IPM's require $O(n^4)$ operations to form the Schur matrix in every iteration.
3. Reoptimization after the addition of cutting planes using a dual simplex variant.

Other approaches

- Solve SDP as a nonlinear program: Burer-Monterio-Zhang.
- Solve SDP as an eigenvalue problem: ACCPM: Oskoorouchi-Goffin, Spectral Bundle: Helmberg-Rendl,Oustry.
- Solve SDP as a convex program: Vanderbei-Benson.
- Solve SDP as a linear program: Krishnan-Mitchell, Goldfarb.
- Solve Schur system via a CG method: Toh.
- Other approaches: Fukuda et al., Kocvara-Stingl.
- IPMs with warm-start capabilities: Sturm.
- Solve SDP via a simplex method: Pataki (96), Krishnan-Pataki-Zhang (this talk).

Primal simplex method for LP I.

Given a feasible basis B, B^{-1}, and an initial bfs $x_B = B^{-1}b$.

1. Construct a simplex multiplier $y^T = c_B^T B^{-1}$.
2. If $s_N = c_N - N^T y \geq 0$ we are optimal; else choose the nonbasic variable with smallest s_j to enter the basis.
3. Perform the minimum ratio test to determine the leaving basic variable.
4. Obtain the new bfs x_B, update the new basis B, and obtain B^{-1} using the product form of the inverse or fast LU updates.

Primal simplex method for LP II.

- **Block pivoting:** Can bring in more than one nonbasic variable with $d_j < 0$; in this case solve the following LP

$$\begin{array}{lrcl} \text{min} & \sum_j s_j^T \alpha_j & & \\ \text{s.t.} & \sum_j (B^{-1} a_j) \alpha_j & \leq & B^{-1} b \\ & \alpha_j & \geq & 0, \ \forall j \end{array}$$

- Entering nonbasic variables the ones with $\alpha_j > 0$; leaving basic variables the ones with zero slacks.
- Check whether the columns of A in the new set are linearly independent; if so we have our new B; else we need to do a crossover to find a new basis B and the associated extreme pt.

Notation I.

Given

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

- **The subspace $\mathbb{B}_X$:**

$$\mathbb{B}_X = \begin{bmatrix} P_1 & P_2 \end{bmatrix} \begin{bmatrix} M & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} P_1^T \\ P_2^T \end{bmatrix}$$

- **Interpretation:** Moving an $\epsilon > 0$ along some direction or its opposite in $\mathbb{B}_X$ keeps us still in $\mathbb{S}^n$.

Notation II.

- **Tangent space:**

$$\mathbb{T}_X = \begin{bmatrix} P_1 & P_2 \end{bmatrix} \begin{bmatrix} M & W \\ W^T & 0 \end{bmatrix} \begin{bmatrix} P_1^T \\ P_2^T \end{bmatrix}$$

- **Interpretation:** Move an $\epsilon > 0$ along some direction or its opposite in $\mathbb{T}_X$; we are at a distance $O(\epsilon^2)$ from $\mathbb{S}^n$.

- **Null space of constraint set:**

$$\mathbb{N} = \{Y : A_i \bullet Y = 0,\ i = 1, \ldots, m\}$$

BFS in SDP I.

- **BFS in SDP:** A feasible $X = P_1 \Lambda P_1^T$, with $\Lambda \succ 0$ and $P_1 \in \mathbb{R}^{n \times r}$, is a bfs iff

$$\mathbb{B}_X \cap \mathbb{N} \;=\; \{0\},$$

- **Alternate characterization:** X is a bfs iff

$$\{P_1^T A_i P_1, i = 1, \ldots, m\} \;\text{ spans }\; \mathbb{S}^r$$

- **Rank at a bfs:** The rank r at a bfs satisfies the inequality

$$\tfrac{r(r+1)}{2} \;\leq\; m$$

Therefore $r < \sqrt{2m}$, and in our case r is $O(\sqrt{n})$.

BFS in SDP II: Crossover algorithm

Given a feasible $X = P_1 M P_1^T$ with $M \succ 0$ and $P_1 \in \mathbb{R}^{n \times r}$.
In each iteration, do the following:

1. If $\{P_1^T A_i P_1, i = 1, \ldots, m\}$ spans $\mathbb{S}^r$, then stop.
2. Find a nonzero $U \in \mathbb{S}^r$ satisfying
$$(P_1^T A_i P_1) \bullet U = 0, \ i = 1, \ldots, m.$$
3. If $C \bullet (P_1 U P_1^T) > 0$, then set $U = -U$.
Calculate $\alpha^* = \max\{\alpha \mid M + \alpha U \succeq 0\}$ and set
$$X = P_1(M + \alpha^* U)P_1^T = \widehat{P}_1 \widehat{M} \widehat{P}_1{}^T$$
where $\widehat{M} \succ 0$, and $\widehat{P}_1 \in \mathbb{R}^{n \times \widehat{r}}$. Let $P_1 = \widehat{P}_1$, $M = \widehat{M}$ and $r = \widehat{r}$. Return to step 1.

Nondegeneracy in SDP

- **Nondegeneracy in SDP:** A feasible X is nondegenerate iff

$$\mathbb{T}_X + \mathbb{N} \; = \; \mathbb{S}^n$$

- **Alternate characterization:** A feasible X in the usual notation is nondegenerate iff

$$B_k \; = \; \begin{bmatrix} P_1^T A_k P_1 & P_1^T A_k P_2 \\ P_2^T A_k P_1 & 0 \end{bmatrix}, \quad k = 1, \ldots, m$$

  are linearly independent in $\mathbb{S}^n$.

- Primal nondegeneracy ⇒ lower bound on the rank of X, i.e.,

$$h(r) \; \geq \; m$$

  where $h(r) = \frac{r(r+1)}{2} + r(n-r)$.

Constructing a dual solution in SDP I.

- $XS = 0$, where $S = (C - \mathcal{A}^T y)$ requires

$$\begin{array}{rcccl} \bar{S}_{11} & = & P_1^T (C - \mathcal{A}^T y) P_1 & = & 0 \\ \bar{S}_{12} & = & P_1^T (C - \mathcal{A}^T y) P_2 & = & 0 \end{array}$$

  This is a set of $h(r)$ equations in m unknowns which is overdetermined if X is nondegenerate.

- If X is a bfs, then the first set of equations gives $\frac{r(r+1)}{2}$ l.i. equations. This is yet not enough to obtain an unique y.

- If X is nondegenerate, we can bring in $(m - \frac{r(r+1)}{2})$ equations from 2nd set to have m l.i. equations to solve for y.

Constructing a dual solution in SDP II.

Consider a simple instance with $n = 8$, $m = 11$, $r = 3$.

$$\left(\begin{array}{c|c} \bar{S}_{11} & \bar{S}_{12} \\ \hline \bar{S}_{12}^{\,T} & \bar{S}_{22} \end{array}\right) = \left(\begin{array}{c|c} P_1^T S P_1 & P_1^T S P_2 \\ \hline P_2^T S P_1 & P_2^T S P_2 \end{array}\right)$$

$$= \left(\begin{array}{ccc|cccccc} \times & \times & \times & \times & \times & \vdots & \circ & \circ & \circ \\ \times & \times & \times & \times & \times & \vdots & \circ & \circ & \circ \\ \times & \times & \times & \times & \circ & \vdots & \circ & \circ & \circ \\ \hline \times & \times & \times & \circ & \circ & \vdots & \circ & \circ & \circ \\ \times & \times & \circ & \circ & \circ & \vdots & \circ & \circ & \circ \\ \cdots & \cdots & \cdots & \cdots & \cdots & . & & & \\ \circ & \circ & \circ & \circ & \circ & & \circ & \circ & \circ \\ \circ & \circ & \circ & \circ & \circ & & \circ & \circ & \circ \\ \circ & \circ & \circ & \circ & \circ & & \circ & \circ & \circ \end{array}\right)$$

Constructing a dual solution in SDP III.

- The first $\frac{r(r+1)}{2}$ equations are chosen from $\bar{S}_{11} = 0$. These are linearly independent.
- Pivot on

$$\begin{bmatrix} \mathrm{svec}(P_1^T A_1 P_1) & \ldots & \mathrm{svec}(P_1^T A_m P_1) \\ \mathrm{vec}(P_1^T A_1 P_2) & \ldots & \mathrm{vec}(P_1^T A_m P_2) \end{bmatrix}$$

until one has m linearly independent equations. Solve the resulting square system of size m for y^*.

- We also considered solving the following LP for y^*

$$\begin{array}{lll} \min_y & \|P_1^T(C - A^T y)P_2\|_1 & \\ \text{s.t.} & \sum_{i=1}^m y_i(P_1^T A_i P_1) & = & P_1^T C P_1 \end{array}$$

The additional equations are distributed over the (1,2) block.

Entering nonbasic variable in SDP

- **LP case:**

  Most negative reduced cost (Dantzig): $\min_{i \in N} s_i$.

  Steepest edge rule (Goldfarb-Reid): $\min_{i \in N} \frac{s_i}{\sqrt{1 + ||B^{-1} a_i||^2}}$.

- **SDP case:**

  Most negative reduced cost: $\min_{||d||=1} d^T S d$.

  This is the normalized eigenvector corr. to $\lambda_{min}(S)$.

Simplex algorithm for SDP

Given a nondegenerate extreme point X in the usual notation.

1. Solve for a dual solution y^*.
2. If $S^* = (C - \mathcal{A}^T y^*) \succeq 0$, stop; else update $\bar{P}$ to be all of P_1, and columns in P_2 corr. to the chosen equations in $\bar{S_{12}} = 0$. Also, compute the normalized eigenvector d corr. to $\lambda_{min}(S^*)$, orthogonalize it w.r.t. $\bar{P}$, and add it to $\bar{P}$.
3. Preprocess and solve the following pivoting SDP for V

   $$\begin{array}{rl} \text{min} & (\bar{P}^T C \bar{P}) \bullet V \\ \text{s.t.} & (\bar{P}^T A_i \bar{P}) \bullet V = b_i, \quad i = 1, \ldots, m \\ & V \succeq 0 \end{array}$$

   Let $V = R \Lambda R^T$, with $M \succ 0$, $P_1 = \bar{P} R$, and $X = P_1 \Lambda P_1^T$. Update P_2. If necessary, do a crossover on X, and return to step 1.

Properties of the algorithm

Theorem. *Let $\{(X^k, y^k, S^k)\}$ be a sequence generated by the algorithm. Then for all k, X^k is primal feasible and $X^k \bullet S^k = 0$. At the k-th iteration, one of the following alternative cases arises:*

1. *If it stops in Step 2, then X^k, (y^k, S^k) are optimal solutions to (SDP) and (SDD), respectively.*
2. *Otherwise we obtain X^{k+1} such that $C \bullet X^{k+1} \leq C \bullet X^k$. If X^k is nondegenerate, then in addition $C \bullet X^{k+1} < C \bullet X^k$.*

Conjecture. *If $\{X^k\}$ contains a nondegenerate subsequence, then either the algorithm terminates in a finite number of steps, or all accumulation points of $\{X^k\}$ are optimal solutions to (SDP).*

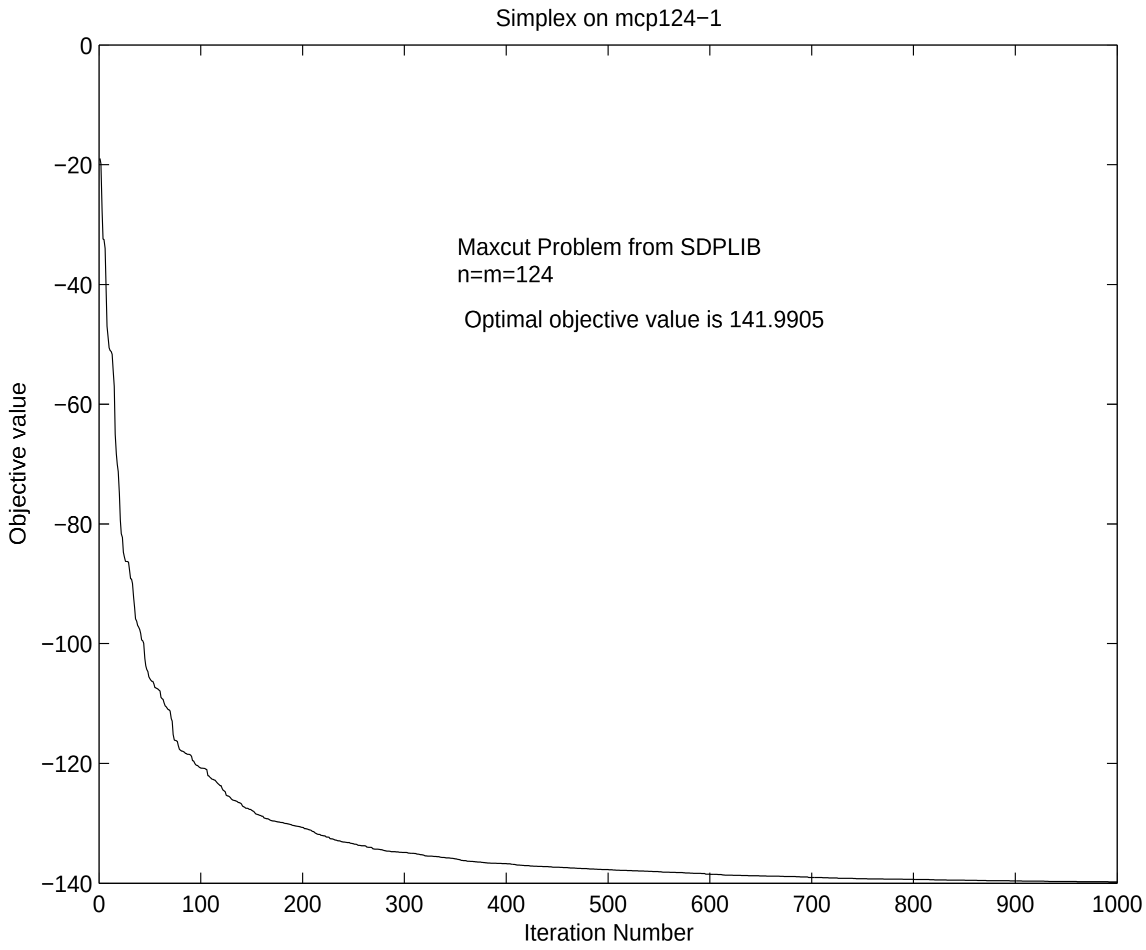
Solving the pivoting SDP

1. Typically the number of constraints m in the pivoting SDP is $O(r^2)$.
2. Let D_i, $i = 1, \ldots, k$ be a basis for the null space of the equality constraints of pivoting SDP.
3. Solve the pivoting SDP as

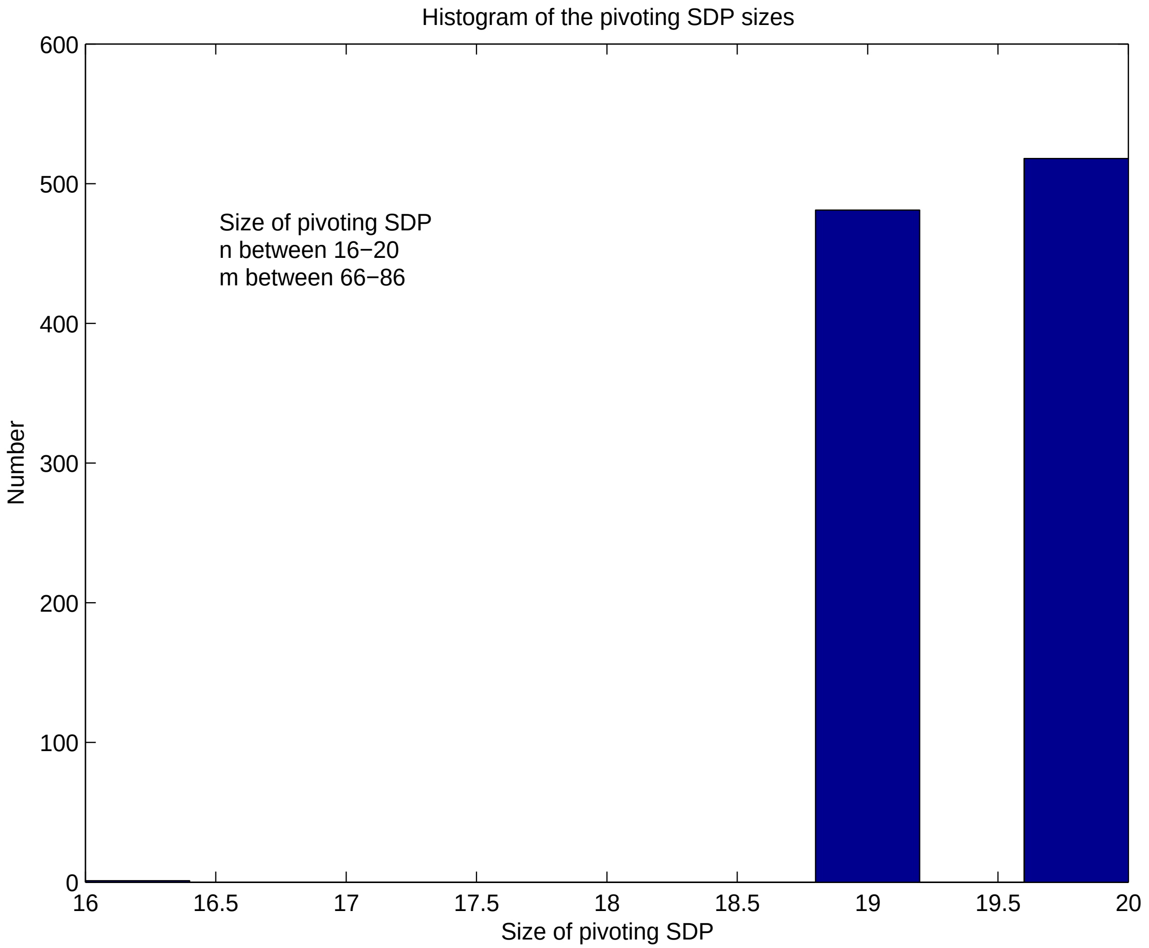
   $$\begin{array}{lrcl} \text{max} & \sum_{i=1}^{k} x_i(\bar{P}^T C\bar{P} \bullet D_i) & & \\ \text{s.t.} & M - \sum_{i=1}^{k} x_i D_i & \succeq & 0 \end{array}$$

   where M is the diagonal matrix of eigenvalues of X.
4. Complexity is dominated by the null space computation, and this is $O(n^3)$, when $m = O(n)$.
5. Is it possible to quickly update the null space, i.e. do this in $O(m^2)$?. (analogous to the basis inverse updates in the simplex method for LP)

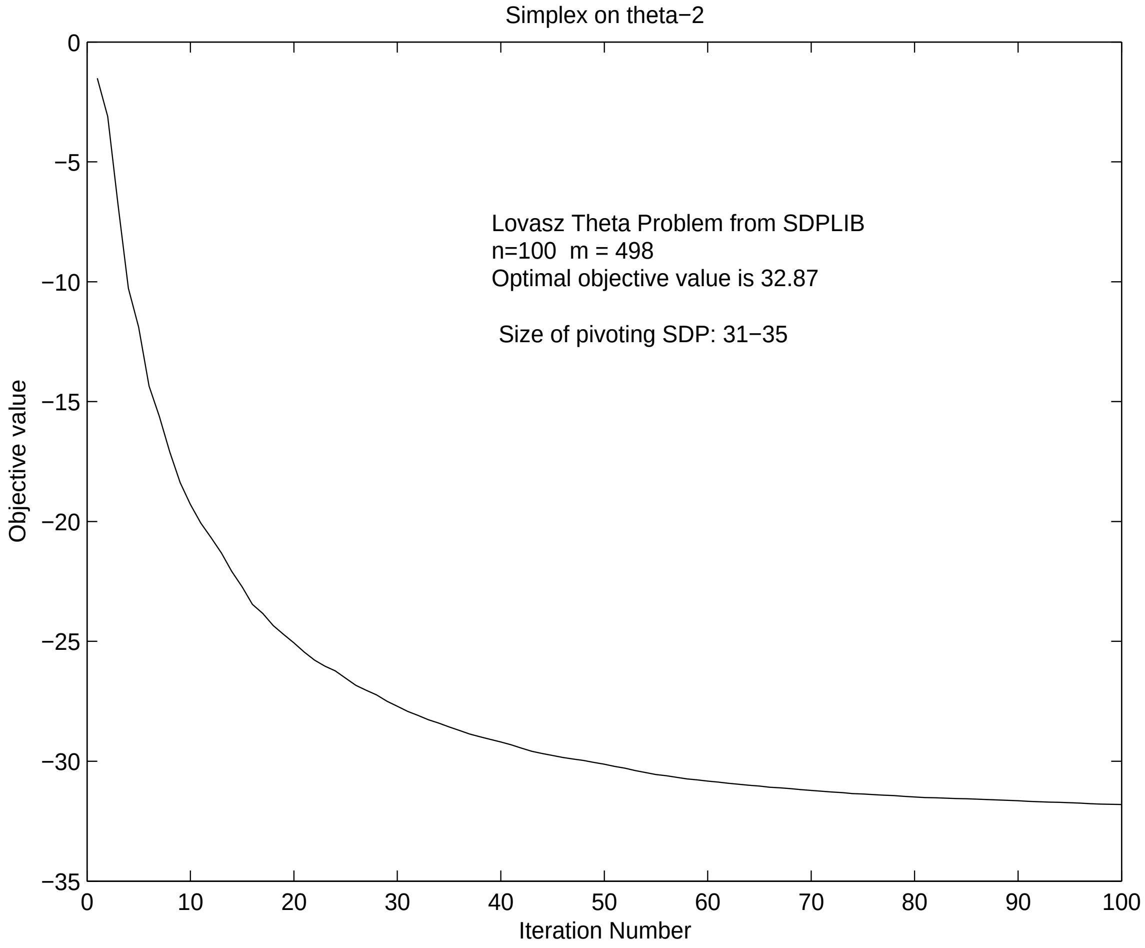
Computational results I.



Computational results II.



Computational results III.



Conclusions and future work

1. A primal active set approach which mimics the primal simplex method for LP.
2. Solve an SDP as a sequence of smaller SDP's in an active set framework. Each iteration carried out in $O(n^3)$ arithmetic operations.
3. We have the general framework for other conic optimization problems including SOCP.
4. Preliminary success with basis updates in SOCP; how about SDP?.
5. Currently investigating dual simplex variants of this method.
6. More computational results in ICCOPT I this August.

References

- F. Alizadeh and D. Goldfarb. Second-order cone programming, Mathematical Programming, 95(2003), pp. 3-51.

- F. Alizadeh, J.P.A. Haeberly, and M.L. Overton. Complementarity and Nondegeneracy in SDP, Mathematical Programming, 77(1997), pp. 111-128.

- B. Borchers. SDPLIB 1.2: A Library of Semidefinite Programming Test Problems, Optimization Methods and Software, 11(1), pp. 683-690, 1999.

- K. Krishnan and J.E. Mitchell. An unifying framework for existing cutting plane methods for semidefinite programming, (to appear in Optimization Methods & Software).

- G. Pataki. Cone-LP's and Semidefinite Programs: Geometry and a Simplex-type Method, Proceedings of the 5th IPCO conference, Springer Verlag 1996.

- G. Pataki. The Geometry of Semidefinite Programming, Handbook of SDP edited by H. Wolkowicz et al., Kluwer Academic Publishers, 2000, pp. 29-65.

- L. Vandenberghe and S. Boyd. Semidefinite Programming, SIAM Review, 38(1996), pp. 49-95.