Towards a practical simplex method for second order cone programming

Kartik Krishnan

AdvOL

Department of Computing and Software
McMaster University
Joint work with Gábor Pataki (UNC), Neha Gupta (IIT Delhi),
and Tamás Terlaky (Mac)

INFORMS Annual Meeting
Denver
Oct 26, 2004

Overview

- Second Order Cone Programming (SOCP)
- Contrast simplex-like approaches and IPMs for conic optimization
- The geometry of SOCP
- Simplex algorithm for SOCP
- Special case: Simplex method for LP
- Properties of the algorithm
- Preliminary computational results
- Conclusions and future work

Second order cone optimization

- **Primal**

$$\begin{array}{rcl} \min \; c_1^T x_1 + c_2^T x_2 + \ldots + c_r^T x_r & & \\ A_1 x_1 + A_2 x_2 + \ldots + A_r x_r & = & b, \\ x_i & \in & \mathcal{K}_i. \end{array} \qquad (SOCP)$$

- **Dual**

$$\begin{array}{rcl} \max \; b^T y & & \\ A_i^T y + s_i & = & c_i, \; i = 1, \ldots, r, \\ s_i & \in & \mathcal{K}_i. \end{array} \qquad (SOCD)$$

- **Notation**

  - $A = (A_1, A_2, \ldots, A_r) \in \mathbb{R}^{m \times n}$ with full row rank.
  - $\mathcal{K} = \mathcal{K}_1 \times \ldots \times \mathcal{K}_r$.
  - Each $\mathcal{K}_i = \left\{ x \in \mathbb{R}^{n_i} : x_1 \geq \left\| x_{2:n_i} \right\| \right\}$ is a second order cone of size n_i, $i = 1, \ldots, r$.

Optimality conditions

- (x^*, y^*, s^*) are optimal iff

$$\begin{aligned} Ax^* = b, \quad x^* \in \mathcal{K}, \qquad & (PF) \\ A^T y^* + s^* = c, \quad s^* \in \mathcal{K}^*, \qquad & (DF) \\ x_i^* \circ s_i^* = 0, \; i = 1, \ldots, r. \qquad & (CS) \end{aligned}$$

- For an SOCP cone in $\mathbb{R}^n$

$$x \circ s \;=\; \begin{pmatrix} x^T s \\ x(1)s(2:n) + s(1)x(2:n) \end{pmatrix}.$$

- For any cone $\mathcal{K}$

$$x \in \mathcal{K},\; s \in \mathcal{K}^*,\; x^T s = 0 \;\Rightarrow\; x \circ s = 0.$$

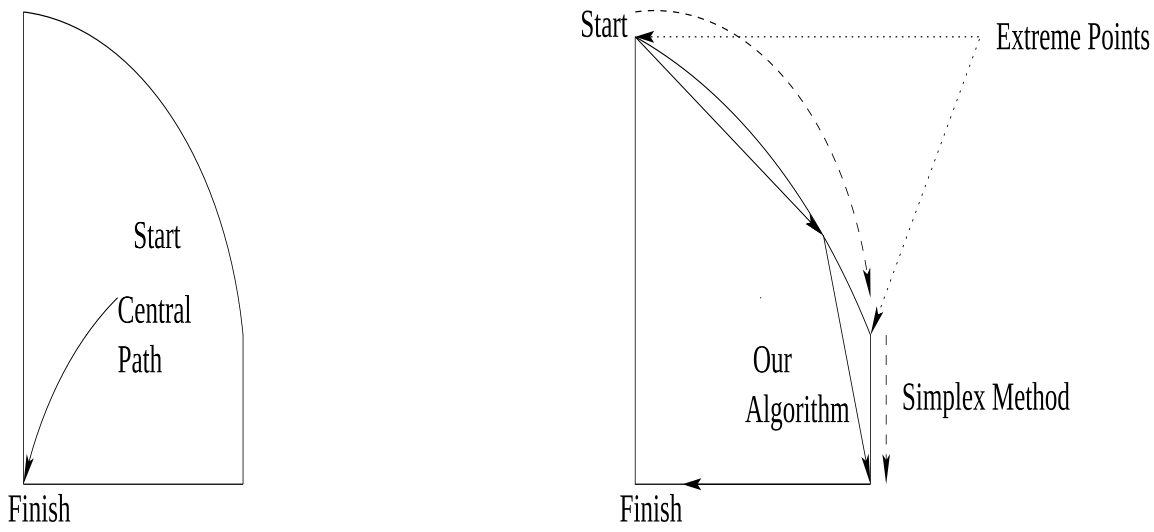
Eigenvalues and Eigenvectors

- Given $\bar{x} \in \mathbb{R}^n$, we have

$$\begin{aligned} \bar{x} &= \tfrac{1}{2}(\bar{x}_1 - ||\bar{x}_{2:n}||)\left(1; -\frac{\bar{x}_{2:n}}{||\bar{x}_{2:n}||}\right) + \tfrac{1}{2}(\bar{x}_1 + ||\bar{x}_{2:n}||)\left(1; \frac{\bar{x}_{2:n}}{||\bar{x}_{2:n}||}\right) \\ &= \lambda_{min}(\bar{x}) v_{min}(\bar{x}) + \lambda_{max}(\bar{x}) v_{max}(\bar{x}). \end{aligned}$$

- **Index classification:** Given $\bar{x}_i \in \mathcal{K}_i$ we have
  - $i \in O$ (zero blocks) if $\lambda_{max}(\bar{x}_i) = 0$.
  - $i \in R$ (boundary blocks) if $\lambda_{min}(\bar{x}_i) = 0$.
  - $i \in I$ (interior blocks) if $\lambda_{min}(\bar{x}_i) > 0$.

Contrast simplex and IPMs for CP I.



Interior Point Method

Feasible Direction 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 important open problem in conic programming.
2. Warm start after branching or the addition of cutting planes using the dual simplex method.
3. It is possible to do every simplex iteration more quickly than an IPM iteration using fast basis LU updates to factorize the basis matrix.

Terminology I.

Given a closed convex cone $K \subset \mathbb{R}^n$. Consider $\bar{x} \in K$

- **Lineality space:**

$$\mathbb{B}_{\bar{x}} \; = \; \{d \in \mathbb{R}^n : \bar{x} \pm \epsilon d \in K, \epsilon > 0\}\,.$$

- **Tangent space:**

$$\mathbb{T}_{\bar{x}} \; = \; \left\{d \in \mathbb{R}^n : \mathrm{dist}(\bar{x} \pm \epsilon d, K) = O(\epsilon^2), \epsilon > 0\right\}.$$

- **Residual space:**

$$\mathbb{R}_{\bar{x}} \; = \; \mathbb{T}_{\bar{x}} \setminus \mathbb{B}_{\bar{x}}.$$

Terminology II.

- **Cone of feasible directions:**

$$\begin{aligned} \mathrm{dir}(\bar{x}, K) &= (K + \mathbb{B}_{\bar{x}}) \\ &= \{d \in \mathbb{R}^n : \bar{x} + \epsilon d \in K, \epsilon > 0\}\,. \end{aligned}$$

- **Tangent cone:**

$$\begin{aligned} \mathbb{TC}(\bar{x}, K) &= \mathrm{cl}(\mathrm{dir}(\bar{x}, K)) \\ &= \left\{d \in \mathbb{R}^n : \mathrm{dist}(\bar{x} + \epsilon d, K) = O(\epsilon^2), \epsilon > 0\right\}. \end{aligned}$$

- **Null space:**

$$\mathbb{N} = \{x \in \mathbb{R}^n : Ax = 0\}\,.$$

Demo

Given $\bar{x}_i \in \mathbb{R}^{n_i}$ on the boundary of the SOCP cone $\mathcal{K}_i$.

- $\mathbb{B}_{\bar{x}_i} = \alpha \bar{x}_i$ for $\alpha \in \mathbb{R}$.
- $\mathbb{T}_{\bar{x}_i} = \mathrm{lin}\left(\mathbb{B}_{\bar{x}_i} \cup \left\{ \begin{pmatrix} 0 \\ w \end{pmatrix} : w^T \bar{x}_{2:n_i} = 0 \right\} \right)$.
- Note $\bar{x}_i + \epsilon \begin{pmatrix} 0 \\ w \end{pmatrix} \notin \mathcal{K}_i$.
- However $\bar{x}_i + \epsilon \begin{pmatrix} 0 \\ w \end{pmatrix} + \epsilon^2 \begin{pmatrix} 1 \\ 0 \end{pmatrix} \in \mathrm{int}(\mathcal{K}_i)$.

Notions of nondegeneracy

A feasible $\bar{x}$ in (SOCP) is

- **c-nondegenerate**

$$\mathbb{T}_{\bar{x}} + \mathbb{N} \;=\; \mathbb{R}^n.$$

  This is a generic property.

- **f-nondegenerate**

$$\mathbb{B}_{\bar{x}} + \mathbb{N} \;=\; \mathbb{R}^n.$$

- **Extreme point**

$$\mathbb{B}_{\bar{x}} \cap \mathbb{N} \;=\; \{0\}.$$

Simplex algorithm for SOCP I.

Given a feasible c-nondegenerate $\bar{x} = (\bar{x}_I; \bar{x}_R; \bar{x}_O)$ in (SOCP). Indices $i \in I$ and $j \in R$ are ranked on $\lambda_{min}(\bar{x}_i)$ and $\bar{x}_{j1}$ respectively.

- 1. **Select basis:** Construct the following basis elements:
  - Decompose $\bar{x} = M_B \bar{x}_B + M_N \bar{x}_N$, where $M_B \in \mathbb{R}^{n \times m}$ and $M_N \in \mathbb{R}^{n \times (n-m)}$. M_B is chosen with Range$(M_B) \subseteq \mathbb{T}_{\bar{x}}$ such that $A_B = AM_B$ is a nonsingular *basis* matrix of size m. Also, $c_B = M_B^T c$.
  - Let B,N be the index set of cones $\mathcal{K}_i$,$i = 1, \ldots, r$ with columns in M_B and M_N respectively. (**Note:** $B \cap N \neq \phi$).
- 2. **Construct dual solution:**
  - Solve $A_B^T y = c_B$ for $\bar{y}$.
  - Compute $\bar{s}_i = c_i - A_i^T \bar{y}_i$, $i \in N$. (**Note:** $\bar{s}_i = 0$, $i \notin N$).

Simplex algorithm for SOCP II.

- 3. **Pricing:** Compute

$$\begin{aligned}\alpha &= \max\{-\lambda_{min}(\bar{s}_i) : \bar{s}_i \notin \mathcal{K}_i, i \in N\}.\\ \beta &= \max\{\bar{x}_i^T \bar{s}_i : \bar{s}_i \in \mathcal{K}_i, i \in N\}.\end{aligned}$$

If $\alpha = \beta = 0$ STOP; $(\bar{x}, \bar{y}, \bar{s})$ is an optimal solution. Else

$$k = \begin{cases} \text{an index s.t. } \alpha = -\lambda_{min}(\bar{s}_k) \text{ if } \alpha > \beta. \\ \text{an index s.t. } \beta = \bar{x}_k^T \bar{s}_k \text{ if } \alpha < \beta. \end{cases}$$

- 4. **Find improving direction:**

  - **Improving component:** Compute $\bar{d}_N \in \mathbb{R}^n$ where

$$\bar{d}_{N_i} = \begin{cases} \min\limits_{||d||=1 \;\; d \in \mathbb{TC}(\bar{x}_k, \mathcal{K}_k)} \bar{s}_k^T d \quad (IC) & \text{if } i = k \\ 0 & \text{for } i = 1, \ldots, r \text{ and } i \neq k. \end{cases}$$

(**Note:** $\text{dir}(\bar{x}_k, \mathcal{K}_k)$ is not CLOSED!)

Simplex algorithm for SOCP III.

- 4. **Find improving direction (continued):**
  - **Centering component:** Construct $\bar{d}_C \in \mathbb{R}^n$ satisfying

$$\bar{d}_{C_i} = \begin{cases} (\alpha; 0) & i \in R \cap B, \text{ a nonzero subset of } \mathbb{R}_{\bar{x}_i} \in \text{Range}(A_B) \\ (\alpha; 0) & i = k \text{ if } k \in R \\ 0 & \text{otherwise} \end{cases}$$

    with $\alpha > 0$ chosen so that $\bar{s}_k^T(\bar{d}_N + \bar{d}_C) < 0$.
  - **Basic component:** Compute $\bar{d}_B \in \mathbb{R}^m$ by solving $A_B d_B = -(A d_N + A d_C)$.
  - The improving direction $\bar{d} = (M_B \bar{d}_B + \bar{d}_C + \bar{d}_N)$.

Simplex algorithm for SOCP IV.

- 5. **Line search:**
  - Compute $\bar{\alpha}$ where

$$\bar{\alpha} \;=\; \max\{\alpha_i : \bar{x}_i + \alpha_i \bar{d}_i \in \mathcal{K}_i, i = 1, \ldots, r\}.$$

  - If $\bar{\alpha} = \infty$ the primal is unbounded. STOP.
  - Else set $\bar{x} = \bar{x} + \bar{\alpha}\bar{d}$ and return to step 1.
    (**Note:** The new $\bar{x}$ is assumed to be c-nondegenerate).

Computing the improving component I.

The solution $\bar{d}_{N_k}$ to (IC) is

- **If** $k \in O$**:**

$$\bar{d}_{N_k} = \begin{cases} -\frac{\bar{s}_k}{||\bar{s}_k||} & \text{if } \lambda_{max}(\bar{s}_k) < 0 \\ v_{min}(\bar{s}_k) & \text{if } \lambda_{min}(\bar{s}_k) < 0 \text{ and } \lambda_{max}(\bar{s}_k) \geq 0 \\ 0 & \text{if } \bar{s}_k \in \mathcal{K}_k \end{cases}$$

- **If** $k \in I$**:**

$$\bar{d}_{N_k} = \begin{cases} -\frac{\bar{s}_k}{||\bar{s}_k||} & \text{if } \bar{s}_k \neq 0 \\ 0 & \text{if } \bar{s}_k = 0 \end{cases}$$

Computing the improving component II.

- **If $k \in R$:** We have

$$\mathbb{TC}(\bar{x}_k, \mathcal{K}_k) \;=\; \{d \in \mathbb{R}^{n_k} : \bar{x}_1 d_1 - \textstyle\sum_{j=2}^{n_k} \bar{x}_j d_j \geq 0\}$$

- In this case $\bar{d}_{N_k}$ is

$$\bar{d}_{N_k} \;=\; \begin{cases} 0 & \text{if } \bar{s}_k = (\bar{x}_1; -\bar{x}_{2:n_k}) \\ \min\limits_{||d||=1,\; d\in\mathbb{TC}(\bar{x}_k,\mathcal{K}_k)} \bar{s}_k^T d & \text{otherwise} \end{cases}$$

Special case: Simplex method for LP

1. The initial iterate is an nondegenerate extreme point solution. If this iterate is not an extreme point, the method is instead a feasible direction method.

2. Given a non-degenerate extreme point, the simplex method chooses the basis matrix as

$$A_B \;=\; A(:,\text{support}(\bar{x})).$$

   This is nonsingular with Range$(A_B) = \mathbb{B}_{\bar{x}}$.

3. For the improving direction $\bar{d}_N = e_k$ (k is the index for which $\bar{s}_i$ is the most negative) and $\bar{d}_B$ the solution to $A_B\bar{d}_B = -A_k$. No centering term d_C is needed. The resulting direction $\bar{d}$ is along an edge of the feasible set.

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. 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 primal and dual optimal solutions to (SOCP) and (SOCD) respectively.*
2. *If it stops in Step 5, then (SOCP) is unbounded and (SOCD) is infeasible.*
3. *Otherwise, if x^k is also c-nondegenerate we have $c^T x^{k+1} < c^T x^k$.*

Convergence of the algorithm

Some practical issues affecting convergence include:

- One also adds a centering term $\bar{d}_{Ci}$ to the cones $i \in I$ which are very nearly in R. This is to prevent the algorithm from getting *jammed* at a suboptimal point.
- There is also the issue of *zigzagging* with the algorithm.

Conjecture. *If $\{x^k\}$ contains a nondegenerate subsequence, then the algorithm with the anti-jamming safeguard either terminates in a finite number of steps, or all accumulation points of $\{x^k\}$ are optimal solutions to (SOCP).*

Comments on the simplex algorithm

- The simplex algorithm for SOCP is a *feasible direction* method that generates search directions in the tangent cone. This direction is suitably centered so as to generate a feasible direction.
- The simplex iterates are not always extreme points and the search directions may traverse the interior of the feasible region.
- The method resembles the *convex simplex* method of Zangwill for minimizing a convex function over a polyhedron.
- The algorithm maintains primal feasibility in every iteration while dual feasibility and complementary slackness are attained at optimality.
- The step length calculation in Step 5 has a closed analytic expression.

Preliminary computational results

Prob	m	n	lp	r	Opt	Obj(0)	Obj(iter)	Iter
hs51	14	22	10	[12]	-5.99	-1.78	-5.99(*)	10
slp1	30	50	20	10(3)	52.83	86.01	52.83	200
slp2	50	130	100	10(3)	14.61	1669.43	14.61	200
slp3	50	160	100	[50 10]	38.51	1773.46	38.51(*)	183
slp4	100	180	150	10(3)	426.64	681.72	426.64(*)	896
slp5	50	80	50	10(3)	45.67	103.66	45.67(*)	153
slp6	50	75	25	10(5)	24.05	85.19	24.17	500
slp7	50	150	100	10(5)	21.27	109.21	21.41	1000
slp8	50	110	100	[10]	82.10	300.68	82.10(*)	157
slp9	100	160	150	[10]	16.92	565.37	16.92(*)	128
slp10	100	300	150	[50 50 50]	705.79	2511.93	715.99	5000
slp11	204	279	147	[56 3 73]	219.93	500.45	222.65	10000

Conclusions and future work

1. A primal simplex approach for conic optimization of which the primal simplex method for LP is a special case.
2. The simplex approach exploits the well known facial structure of SOCP problems developed in Pataki (2000).
3. We have the framework for solving conic optimization problems over LP, SOCP and SDP cones.
4. We also have a dual simplex variant which mimics the dual simplex method for LP.
5. Currently investigating fast basis inverse (LU) updates to speed up the algorithm.
6. Future use in warm start after branching or the addition of cutting planes.

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.

- M.S. Bazaraa, H.D. Sherali and C.M. Shetty. Nonlinear Programming: Theory and Algorithms, 2nd edition, John Wiley 1993.

- K. Krishnan, G. Pataki, N. Gupta and T. Terlaky. A feasible direction method for SOCP, forthcoming.

- I. Maros. Computational techniques of the simplex method. Kluwer International Series, 2003.
- 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.