A Note on the Simplex Method for 2-Dimensional Second-Order Cone Programming

Yu Xia[*]

The Institute of Statistical Mathematics,
4-6-7 Minami-Azabu, Minato-Ku,
Tokyo 106-8569, Japan
yuxia@ism.ac.jp

Abstract. We transform a 2-dimensional second-order cone program to a standard linear program in order to pivot a vector based on its three states. Related properties of the transformation are given. Based on the transformation, we interpret the simplex method and its sensitivity analysis for the 2-dimensional second-order cone programming, especially the state changes. Finally, we give some applications of the 2-dimensional second-order cone programming.

Keywords: Second-order cone programming, linear programming, simplex method, basis, sensitivity analysis.

1 Introduction

A second-order cone (SOC) in $\mathbb{R}^{n+1}$ is the set

$$\mathcal{Q}_{n+1} \overset{\text{def}}{=} \left\{ \mathbf{x} \in \mathbb{R}^{n+1} \colon x_0 \geq \sqrt{\sum_{i=1}^{n} x_i^2} \right\} .$$

We omit the subscript $n+1$ of $\mathcal{Q}$ when it is clear from the context. We write $\mathbf{x} \geq_{\mathcal{Q}} \mathbf{0}$ interchangeably with $\mathbf{x} \in \mathcal{Q}$, since $\mathcal{Q}$ induces a partial order. Other names of $\mathcal{Q}$ include Loréntz cone, ice-cream cone, and quadratic cone.

The standard form primal and dual second-order cone programming (SOCP) problem is the following:

Primal   **Dual**
min $\mathbf{c}_1^T\mathbf{x} + \cdots + \mathbf{c}_n^T\mathbf{x}$ max $\mathbf{b}^T\mathbf{y}$
s.t. $A_1\mathbf{x}_1 + \cdots + A_n\mathbf{x}_n = \mathbf{b}$,   s.t. $A_i^T\mathbf{y}_i + \mathbf{s}_i = \mathbf{c}_i\ (i=1,\ldots,n)$,   (1)
$\mathbf{x}_i \geq_{\mathcal{Q}_{N_i}} \mathbf{0}\ (i=1,\ldots,n)$; $\mathbf{s}_i \geq_{\mathcal{Q}_{N_i}} \mathbf{0}\quad (i=1,\ldots,n)$.

Here $N_i \in \mathbb{N}$ is the dimension of variables $\mathbf{x}_i$ and $\mathbf{s}_i$ $(i = 1, \ldots, n)$; $\mathbf{y} \in \mathbb{R}^m$ is a dual variable; $\mathbf{b} \in \mathbb{R}^m$, $\mathbf{c}_i \in \mathbb{R}^{N_i}$, $A_i \in \mathbb{R}^{m \times N_i}$ $(i = 1, \ldots, n)$ are data.

SOCP is a useful tool for many practical applications and theoretical developments; see [1], [2] for a survey. Interior point methods have been extended to SOCP. On the other hand, simplex methods remain widely used practical procedures for linear programming (LP) because of the low running time of each step – linear vs cubic for that of interior point methods. These features are especially useful for " warm starting " and large scale computing. Parallel simplex methods and cluster computing grids further facilitate the very large scale application. Unfortunately, no simplex-like method possessed of the above merits exists for SOCP, because of the nonlinear nature of SOC. In this note, we consider a subclass of SOCP which simplex methods can be extended to. Observe that an LP model is a 1-dimensional SOCP model. When $N_i = 2 (i = 1, \ldots, n)$, (1) is the following LP model, where we write the jth entry of $\mathbf{x}_i$ as $(x_i)_j$.

Primal		**Dual**	
min	$\mathbf{c}_1^T \mathbf{x} + \cdots + \mathbf{c}_n^T \mathbf{x}$	max	$\mathbf{b}^T \mathbf{y}$
s.t.	$A_1 \mathbf{x}_1 + \cdots + A_n \mathbf{x}_n = \mathbf{b}$,	s.t.	$A_i^T \mathbf{y}_i + \mathbf{s}_i = \mathbf{c}_i$ $(i = 1, \ldots, n)$,
	$(x_i)_0 \geq (x_i)_1$ $(i = 1, \ldots, n)$,		$(s_i)_0 \geq (s_i)_1$ $(i = 1, \ldots, n)$,
	$(x_i)_0 \geq -(x_i)_1$ $(i = 1, \ldots, n)$;		$(s_i)_0 \geq -(s_i)_1$ $(i = 1, \ldots, n)$.

$$(2)$$

Note that the constraints $(x_i)_0 \geq 0 \, (i = 1, \ldots, n)$ can be obtained by adding $(x_i)_0 - (x_i)_1 \geq 0$ and $(x_i)_0 + (x_i)_1 \geq 0$ together. Nevertheless, it is not straightforward to extend the pivot rules for the simplex method to (2). Pivots for LP change variable states between positivity and zero — a variable enters the basis when it becomes zero and leaves the basis when it becomes positive. However, a vector associated with an SOC has three states: in the interior of the SOC (int $\mathcal{Q}$), in the boundary of the SOC (bd $\mathcal{Q}$), and zero.

In this note, we show how to transform (2) into a standard form LP model. With this, we interpret the simplex method and its sensitivity analysis for the 2-dimensional SOCP based on the original SOC variables of (2). We also give some applications of the 2-dimensional SOCP for motivations of developing efficient algorithms for it.

The rest of the paper is organized as follows. In § 2, we give a transformation that maps (2) into a standard form LP model. Properties of the transformation are also discussed. In § 3, we interpret the simplex method for the 2-dimensional SOCP and give the sensitivity analysis for the simplex method. In § 4, we present some applications of the 2-dimensional SOCP.

2   The Transformation

In this part, we give a transformation of a 2-dimensional second-order cone program to a standard form linear program. We also prove the equivalence of the standard form linear program to (2) and give its properties.

First, we observe that for any $\mathbf{x}_i \in \mathcal{Q}_1 (i \in \{1, \ldots, n\})$, the state of $\mathbf{x}_i$ is determined by the number of active constraints:

$$(x_i)_0 + (x_i)_1 \geq 0 , \qquad\qquad ((+)\text{-constraint})$$
$$(x_i)_0 - (x_i)_1 \geq 0 . \qquad\qquad ((-)\text{-constraint})$$

That is, $\mathbf{x}_i = \mathbf{0}$ iff both $(+)$ and $(-)$-constraints are active; $\mathbf{x}_i \in \mathrm{bd}\,\mathcal{Q}$ iff only one of the $(+)$ and $(-)$ constraints is active; $\mathbf{x}_i \in \mathrm{int}\,\mathcal{Q}$ iff neither of the $(+1)$ and $(-)$ constraints is active.

The transformation. Define

$$\tilde{P} \overset{\text{def}}{=} \begin{pmatrix} \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} \end{pmatrix} .$$

Then $\tilde{P}$ is an orthogonal matrix. In addition, $\tilde{P}^T = \tilde{P}$, $\tilde{P}^2 = I$.

Let $\mathbf{v}_i = [(v_i)_0, (v_i)_1]^T \overset{\text{def}}{=} \tilde{P}\mathbf{x}_i \ (i = 1, \ldots, n)$. Then the following holds.

1. $\mathbf{x}_i \geq_{\mathcal{Q}} \mathbf{0}$ iff $\mathbf{v}_i \geq \mathbf{0}$.
   In addition, $\mathbf{x}_i \in \mathrm{int}\,\mathcal{Q}$ iff $\mathbf{v}_i > \mathbf{0}$; $\mathbf{x}_i = \mathbf{0}$ iff $\mathbf{v}_i = \mathbf{0}$; $\mathbf{x}_i \in \mathrm{bd}\,\mathcal{Q}$ iff one of $(v_i)_0$ and $(v_i)_1$ is positive and the other one is zero.
2. $|(v_i)_0|$ is the distance of $\mathbf{x}_i$ from the line $(x_i)_0 = -(x_i)_1$; $|(v_i)_1|$ is the distance of $\mathbf{x}_i$ from the line $(x_i)_0 = (x_i)_1$; $\|\mathbf{v}_i\|_2$ is the distance of $\mathbf{x}_i$ from $\mathbf{0}$.
   (a) $\mathbf{v}_i \geq \mathbf{0} \Longleftrightarrow \mathbf{x}_i \in \mathcal{Q}$ and $\mathbf{v}_i$ are the distances of $\mathbf{x}_i$ from the two boundaries of SOC, i.e. the $(+)$ and $(-)$-constraints.
   (b) $(v_i)_0 \geq 0$ and $(v_i)_1 < 0 \Longleftrightarrow \mathbf{x}_i \notin \mathcal{Q}$ and $-(v_i)_1$ is the distance of $\mathbf{x}_i$ from SOC.
   (c) $(v_i)_0 < 0$ and $(v_i)_1 \geq 0 \Longleftrightarrow \mathbf{x}_i \notin \mathcal{Q}$ and $-(v_i)_0$ is the distance of $\mathbf{x}_i$ from SOC.
   (d) $\mathbf{v}_i \leq \mathbf{0} \Longleftrightarrow \mathbf{x}_i \notin \mathcal{Q}$ and $\|\mathbf{v}_i\|_2$ is the distance of $\mathbf{x}_i$ from SOC.
3. $\sqrt{2}\mathbf{v}_i$ are the two eigenvalues of $\mathbf{x}_i$. The columns of $\frac{\sqrt{2}}{2}\tilde{P}$ are two eigenvectors of $\mathbf{x}_i$. In other words, $\left(\frac{\sqrt{2}}{2}\tilde{P}\right)(\sqrt{2}\mathbf{v}_i)$ is the spectral decomposition of $\mathbf{x}_i$ (see [1]).

Standard form linear program. Let P be a block diagonal matrix with each block being $\tilde{P}$. Let $\tilde{\mathbf{c}} \overset{\text{def}}{=} P\mathbf{c}$, $\tilde{A} \overset{\text{def}}{=} AP$. We consider the following standard form linear program:

	Primal	**Dual**	
	min $\tilde{\mathbf{c}}^T\mathbf{v}$	max $\mathbf{b}^T\mathbf{y}$	
s.t.	$\tilde{A}\mathbf{v} = \mathbf{b}$,	s.t. $\tilde{A}^T\mathbf{y} + \mathbf{w} = \tilde{\mathbf{c}}$,	(3)
	$\mathbf{v} \geq \mathbf{0}$;	$\mathbf{w} \geq \mathbf{0}$.	

Complementary slackness. Let $(\mathbf{v}_i; \mathbf{w}_i) = (P\mathbf{x}; P\mathbf{s})$. Then

$$\mathbf{v}_i \cdot \mathbf{w}_i = \mathbf{0} \Longleftrightarrow \mathbf{x}_i \circ \mathbf{s}_i = \mathbf{0} , \qquad (4)$$

where $\mathbf{v}_i \cdot \mathbf{w}_i \overset{\text{def}}{=} \begin{bmatrix} (v_i)_0(w_i)_0 \\ (v_i)_1(w_i)_1 \end{bmatrix}$, $\mathbf{x}_i \circ \mathbf{s}_i \overset{\text{def}}{=} \begin{bmatrix} (x_i)_0(s_i)_0 + (x_i)_1(s_i)_1 \\ (x_i)_0(s_i)_1 + (x_i)_1(s_i)_0 \end{bmatrix} .$

Equivalence between (2) *and* (3). By duality theorem, a triple $(\mathbf{v}; \mathbf{w}; \mathbf{y})$ solves (3) iff

$$\tilde{A}\mathbf{v} = \mathbf{b}$$
$$\tilde{A}^T\mathbf{y} + \mathbf{w} = \tilde{\mathbf{c}}$$
$$\mathbf{v} \cdot \mathbf{w} = \mathbf{0}$$
$$\mathbf{v} \geq \mathbf{0}$$
$$\mathbf{w} \geq \mathbf{0} \ .$$

And $(\mathbf{x}; \mathbf{s}; \mathbf{y})$ is a solution to (2) iff it satisfies the following conditions [1].

$$A\mathbf{x} = \mathbf{b}$$
$$A^T\mathbf{y} + \mathbf{s} = \mathbf{c}$$
$$\mathbf{x} \circ \mathbf{s} = \mathbf{0}$$
$$\mathbf{x} \geq_{\mathcal{Q}} \mathbf{0}$$
$$\mathbf{s} \geq_{\mathcal{Q}} \mathbf{0} \ .$$

Together with (4) and the properties of P, we have that $(\mathbf{x}; \mathbf{s}; \mathbf{y})$ is a solution to (2) iff $(P\mathbf{x}; P\mathbf{s}; \mathbf{y})$ solves (3). In addition, they have the same objective value.

Strong duality. The properties of P and (4) also imply that the strong duality for the 2-dimensional SOCP holds if (i) both the primal and the dual have feasible solutions, or (ii) the primal has feasible solutions and the objective value is below bounded in the feasible region. For higher dimensional SOCP, neither (i) nor (ii) is sufficient for strong duality, see [1].

3   The Simplex Method

In this part, we interpret the simplex method for the 2-dimensional SOCP.

Basic solution. Without loss of generality, we assume A has full row rank; otherwise, either the linear constraints are inconsistent or some linear constraints are redundant. Let A_B be an $m \times m$ nonsingular matrix of A. Since P is nonsingular, $\tilde{A}_B$ is nonsingular, too. The constraints for which the column of the corresponding variable $\mathbf{v}_i$ belongs to $\tilde{A}_B$ are the basic constraints. Other constraints are the nonbasic constraints. The set of basic constraints is the basis. The corresponding vector $\mathbf{x}$ is the basic solution. If a basic solution $\mathbf{x}$ also satisfies $A\mathbf{x} = \mathbf{b}$ and $\mathbf{x} \geq_{\mathcal{Q}} \mathbf{0}$, $\mathbf{x}$ is called a basic feasible solution.

Let B_x represent the number of boundary blocks of a basic solution $\mathbf{x}$, I_x the number of its interior blocks, and O_x the number of its zero blocks. Then

$$B_{x^*} + 2I_{x^*} \leq m \ .$$

The simplex method. Next we interpret the primal simplex method with Bland's pivoting rule. Other simplex methods can be explained in a similar way.

1. Solve a phase I problem to get a basic feasible solution to the primal of (3). Assume the corresponding partition of the index set $\{i(k) \mid i \in \{1, \ldots, n\},$ $k \in \{-, +\}\}$ (i is the block index, k indicates the boundary constraints $(+)$ or $(-)$) is B and N, where $\tilde{A}_B \in \mathbb{R}^{m \times m}$ is nonsingular. Let $\mathbf{v}_B$ and $\mathbf{v}_N$ be the basic and nonbasic constraints, i.e. $\mathbf{v}_B = \tilde{A}_B^{-1}\mathbf{b} - \tilde{A}_B^{-1}\tilde{A}_N\mathbf{v}_N$.
2. If $\tilde{\mathbf{c}}_N - (\tilde{A}_B^{-1}\tilde{A}_N)^T\tilde{\mathbf{c}}_B \geq 0$, $\mathbf{x} = P\mathbf{v}$ is optimal for (2). Stop. Otherwise, there exists index $i(k)$ such that $\left(\tilde{\mathbf{c}}_N - (\tilde{A}_B^{-1}\tilde{A}_N)^T\tilde{\mathbf{c}}_B\right)_{i(k)} < 0$. That indicates that if $\mathbf{x}_i$ is moved away from the boundary k, the objective may be decreased.
3. Check the columns of $(\tilde{A}_B^{-1}\tilde{A}_N)_{i(k)}$ for such $i(k)$'s. If there exists an $i(k)$ such that $(\tilde{A}_B^{-1}\tilde{A}_N)_{i(k)} \leq \mathbf{0}$; then the problem is unbounded, i.e. $\mathbf{x}_i$ can be moved arbitrarily away from the boundary k to decrease the objective value infinitely. Otherwise, from the $i(k)$'s choose the smallest index $\bar{i}(\bar{k})$; from the indices $j(l)$'s with $(\tilde{A}_B^{-1}\tilde{A}_N)_{j(l),i(k)} > 0$ choose the smallest index $\bar{j}(\bar{l})$. Move $\mathbf{x}_{\bar{j}}$ to boundary $\bar{l}$, and move $\mathbf{x}_{\bar{i}}$ away from boundary $\bar{k}$ at a distance $(\tilde{A}_B^{-1}\mathbf{b})_{\bar{j}(\bar{l})}/(\tilde{A}_B^{-1}\tilde{A}_N)_{\bar{j}(\bar{l}),\bar{i}(\bar{k})}$.
4. Go to step 2 with the new basic, nonbasic constraints and coefficient matrix.

The state of the variable. In the above algorithm, each pivot affects at most two constraints: one active constraint becomes inactive, and one inactive constraint becomes active. Next, we consider how the state of $\mathbf{x}$ is affected by the pivots.

1. $\bar{i} = \bar{j}$
   In this case, $\mathbf{x}_{\bar{i}}$ must be moved from one boundary to the other boundary of $\mathcal{Q}$.
2. $\bar{i} \neq \bar{j}$
   In this case, the pivot affects two variables. The total number of active constraints for $\mathbf{x}_{\bar{i}}$ and $\mathbf{x}_{\bar{j}}$ is unchanged after the pivot. That total number can be only 1, 2, 3.
   (a) *The number is 1.* This means that the pivot makes an interior variable boundary and a boundary variable interior.
   (b) *The number is 2.* This means that after the pivot, a zero and an interior variable become two boundary variables, or vice versa.
   (c) *The number is 3.* This means that a zero variable is changed to a boundary variable, and a boundary variable is changed to a zero variable by the pivot.

Other methods for linear program, such as dual simplex algorithm, primal-dual simplex algorithm (see [3]) can also be applied to the 2-dimensional second-order cone program.

Sensitivity analysis. We can also perform sensitivity and parameter analysis on the 2-dimensional second-order cone programming.

Given a basic optimal solution for the primal-dual pair (2) with objective value ζ:

$$\zeta = \zeta^* - {\mathbf{w}_N^*}^T \mathbf{v}_N^*$$
$$\mathbf{v}_B = \mathbf{v}_B^* - \tilde{A}_B^{-1} \tilde{A}_N \mathbf{v}_N^* \, ,$$

where

$$\mathbf{v}_B^* = \tilde{A}_B^{-1} \mathbf{b}$$
$$\mathbf{v}_N^* = \mathbf{0}$$
$$\mathbf{w}_N^* = \tilde{A}_N^T \tilde{A}_B^{-T} \tilde{\mathbf{c}}_B - \tilde{\mathbf{c}}_N$$
$$\zeta^* = \tilde{\mathbf{c}}_B^T \tilde{A}_B^{-1} \mathbf{b} \, .$$

Next, we give the ranges of changes in cost coefficients $\mathbf{c}$ and right-hand side $\mathbf{b}$, under which current classification of optimal basic and nonbasic constraints remains optimal.

We first consider the change of right-hand side $\mathbf{b}$.

Assume $\mathbf{b}$ is changed to $\mathbf{b} + t\Delta\mathbf{b}$. The original basic constraints are now

$$\bar{\mathbf{v}}_B = \tilde{A}_B^{-1}(\mathbf{b} + t\Delta\mathbf{b}) \geq \mathbf{0} \, .$$

If $\bar{\mathbf{v}}_B \geq \mathbf{0}$, the current basis is still optimal. Solve $\bar{\mathbf{v}}_B \geq \mathbf{0}$ for t, we obtain

$$\left(\min_{j(l) \in B} -\frac{(\tilde{A}_B^{-1}\Delta\mathbf{b})_{j(l)}}{(\tilde{A}_B^{-1}\mathbf{b})_{j(l)}} \right)^{-1} \leq t \leq \left(\max_{j(l) \in B} -\frac{(\tilde{A}_B^{-1}\Delta\mathbf{b})_{j(l)}}{(\tilde{A}_B^{-1}\mathbf{b})_{j(l)}} \right)^{-1} \, .$$

And the objective will be $\zeta^* + t\tilde{\mathbf{c}}_B^T \tilde{A}_B^{-1} \Delta\mathbf{b}$.

Assume $\mathbf{c}$ is changed to $\mathbf{c} + t\Delta\mathbf{c}$. The dual nonbasic constraints are now

$$\bar{\mathbf{w}}_N = \mathbf{w}^* + \Delta\mathbf{w}_N \, ,$$

where

$$\Delta\mathbf{w}_N = \left(\tilde{A}_B^{-1} \tilde{A}_N \right)^T \Delta\tilde{\mathbf{c}}_B - \Delta\tilde{\mathbf{c}}_N \, .$$

The current basis remains optimal if $\bar{\mathbf{w}}_N \geq \mathbf{0}$. Solve $\bar{\mathbf{w}}_N \geq \mathbf{0}$ for t, we obtain

$$\left(\min_{i(k) \in N} -\frac{\Delta\mathbf{w}_{i(k)}}{\mathbf{w}_{i(k)}} \right)^{-1} \leq t \leq \left(\max_{i(k) \in N} -\frac{\Delta\mathbf{w}_{i(k)}}{\mathbf{w}_{i(k)}} \right)^{-1} \, .$$

And the objective will be $\zeta^* + t\Delta\tilde{\mathbf{c}}_B^T \tilde{A}_B^{-1} \mathbf{b}$.

Combining the above results, we can obtain that for simultaneous changes of right-hand side and cost coefficients.

4   Application

In this part, we give some applications of the 2-dimensional second-order cone programming.

Given a scalar u, its absolute value $|u|$ is equivalent to the optimal value of the following 2-dimensional SOCP model:

$$\min u_0$$
$$\text{s.t. } (u_0, u)^T \geq_\mathcal{Q} \mathbf{0}.$$

Therefore, given $A_i \in \mathbb{R}^{n_i \times m}$, $\mathbf{b}_i \in \mathbb{R}^{n_i}$ $(i = 1, \dots, k)$, the L^1 and L^∞ norms associated with the affine transformation of a vector $A_i \mathbf{x} + \mathbf{b}_i$ can be formed as 2-dimensional SOCP models. Below are some examples.

Minimize the sum of weighted norms. Given some weights w_i $(i = 1, \dots, k)$ for the k affine transformations of a vector $\mathbf{x}$, the problem of minimizing the weighted sum of L^1 or L^∞ norm can be formulated as a 2-dimensional SOCP model.

1. Minimize the sum of weighted L^1- norms.

$$\min \sum_{i=1}^k w_i \sum_{j=1}^{n_i} (u_{ij})_0$$
$$\text{s.t. } A_i \mathbf{x} + \mathbf{b}_i = \mathbf{u}_i \quad (i = 1, \dots, k)$$
$$[(u_{ij})_0, (u_i)_j]^T \geq_\mathcal{Q} \mathbf{0} \quad (i = 1, \dots, k; j = 1, \dots, n_i).$$

   We use $(u_i)_j$ to represent the *jth* entry of vector $\mathbf{u}_i = [(u_i)_1, \dots, (u_i)_{n_i}]^T$. And $(u_{ij})_0$ is a variable not belonging to $\mathbf{u}_i$.

2. Minimize the sum of weighted $L^\infty-$ norms.

$$\min \sum_{i=1}^k w_i (u_i)_0$$
$$\text{s.t. } A_i \mathbf{x} + \mathbf{b}_i = \mathbf{u}_i \quad (i = 1, \dots, k)$$
$$[(u_i)_0, (u_i)_j]^T \geq_\mathcal{Q} \mathbf{0} \quad (i = 1, \dots, k; j = 1, \dots, n_i).$$

Minimize the largest norm. The problem of minimizing the largest L^1 or L^∞ norm of the k norms can be cast as a 2-dimensional SOCP model.

1. Minimize the largest L^1- norm.

$$\min t$$
$$\text{s.t. } A_i \mathbf{x} + \mathbf{b}_i = \mathbf{u}_i \quad (i = 1, \dots, k),$$
$$[(u_{ij})_0, (u_i)_j]^T \geq_\mathcal{Q} \mathbf{0} \quad (i = 1, \dots, k; j = 1, \dots, n_i),$$
$$\left[t, \sum_{j=1}^{n_i} (u_{ij})_0\right]^T \geq_\mathcal{Q} \mathbf{0} \quad (i = 1, \dots, k).$$

2. Minimize the largest $L^\infty-$ norms.
   The problem $\min \max_{1 \leq i \leq k} \|\bar{\mathbf{v}}_i\|_1$ can be formulated as the follow.

$$\min t$$
$$\text{s.t. } A_i \mathbf{x} + \mathbf{b}_i = \mathbf{u}_i \quad (i = 1, \dots, k),$$
$$[t, (u_i)_j]^T \geq_\mathcal{Q} \mathbf{0} \quad (i = 1, \dots, k; j = 1, \dots, n_i).$$

Minimize the sum of r largest norms. Similarly as minimizing the sum of r largest Euclidean norms [1], the problem of minimizing the sum of r largest L^1 or L^∞ norms can be formulated as a 2-dimensional SOCP model.

1. Minimize the sum of r largest L^1- norm.

$$\min \sum_{i=1}^{r} u_i + rt$$
$$\text{s.t.} \ \ A_i\mathbf{x} + \mathbf{b}_i = \mathbf{v}_i \quad (i = 1, \ldots, k)$$
$$[(v_{ij})_0, (v_i)_j]^T \geq_{\mathcal{Q}} \mathbf{0} \quad (i = 1, \ldots, k; \ j = 1, \ldots, n_j)$$
$$\left[t + u_i, \sum_{j=1}^{n_i} (v_{ij})_0\right]^T \geq_{\mathcal{Q}} \mathbf{0} \quad (i = 1, \ldots, k)$$

2. Minimize the sum of r largest $L^\infty-$ norm.

$$\min \sum_{i=1}^{r} u_i + rt$$
$$\text{s.t.} \ \ A_i\mathbf{x} + \mathbf{b}_i = \mathbf{v}_i \quad (i = 1, \ldots, k)$$
$$[(v_i)_0, (v_i)_j]^T \geq_{\mathcal{Q}} \mathbf{0} \quad (i = 1, \ldots, k; \ j = 1, \ldots, n_j)$$
$$[t + u_i, (v_i)_0]^T \geq_{\mathcal{Q}} \mathbf{0} \quad (i = 1, \ldots, k)$$

Here, $\mathbf{v}_i = [(v_i)_1, \ldots, (v_i)_{n_i}]^T$. And t, u_i, $(v_{ij})_0$, $(v_i)_0$ are scalar variables.

5   Conclusion

We've transformed a 2-dimensional SOCP model into a standard form LP model and interpreted the simplex method and its sensitivity analysis for the model. Some applications have also been given.

References

1. F. Alizadeh and D. Goldfarb. Second-order cone programming. *Math. Program.*, 95(1, Ser. B):3–51, 2003.
2. Miguel Sousa Lobo, Lieven Vandenberghe, Stephen Boyd, and Hervé Lebret. Applications of second-order cone programming. *Linear Algebra Appl.*, 284(1-3):193–228, 1998.
3. Robert J. Vanderbei. *Linear programming: foundations and extensions.* Kluwer Academic Publishers, Boston, MA, 1996.