Title:
A Note on 2-dimensional Second-order Cone Programming and Its Simplex Method

Authors:
Farid Alizadeh and Yu Xia

AdvOl-Report No. 2004/16
October 2004, Hamilton, Ontario, Canada
A Note on 2-dimensional Second-order Cone Programming and Its Simplex Method

Farid Alizadeh∗ Yu Xia†

October 9, 2004

1 Introduction

Second-order cone programming is a useful tool for many practical applications and theoretical developments; see [2, 6] for a survey. As a linear program, interior point methods can approximate a second-order cone program in polynomial time. On the other hand, simplex methods remain to be widely used practical procedures for the linear programming because of its “warm start” ability and low running time of each step – linear vs cubic for interior point methods. These features are especially useful for large scale computing. In addition, parallel simplex methods and cluster computing grids further facilitate the very large scale application. Unfortunately, no simplex-like methods that process the above merits exist for second-order cone programming. In this short note, we will give a transformation between the 2-dimensional second-order cone programs and the linear programs and its associated simplex algorithms.

Notations we use are below. Bold lower case letters for column vectors; lower case letters for scalars; capital letters for matrices. Primal and dual vectors (x and s) are indexed from 0; thus, jth entry of vector x_i is written as (x_i)_j. Superscript T is used to represent the transpose of a matrix or vector. We use “;” to concatenate column vectors.

A second-order cone in $\mathbb{R}^{n+1}$ is denoted by

$$Q_{n+1} \overset{\text{def}}{=} \left\{ x \in \mathbb{R}^{n+1} : x_0 \geq \sqrt{\sum_{i=1}^{n} x_i^2} \right\}.$$ 

It is self dual (see [1, 5]). Other names of Q include Loréntz cone, ice-cream cone, and quadratic cone. We write $x \geq Q_{n+1} 0$ interchangeably with $x \in Q_{n+1}$.

We use $e_n(i)$ to denote a vector in $\mathbb{R}^n$ whose ith entry is 1 and the remaining entries are 0. We omit the subscript n, just write $e_i$ when the dimension is clear from the context.

In § 2, we will give a transformation that maps a 2-dimensional second-order cone programming into a standard linear programming. Properties of the transformation will also be given. In § 3, we will interpret the simplex method for the second-order cone programming. Sensitive analysis will also be given. In § 4, we will show some application of the 2-dimensional second-order cone programming.

∗RUTCOR and Business School, Rutgers, the State University of New Jersey, U.S.A. alizadeh@rutcor.rutgers.edu, research supported in part by the U.S. National Science Foundation.

†Computing and Software, McMaster University, 1280 Main Street West, Hamilton, ON L8S 4K1, Canada yuxia@cas.mcmaster.ca
2 The Transformation

2.1 The 2-Dimensional Second-order Cone Programming

As convention, we write a second-order cone programming (SOCP) problem in the following standard primal and dual pair:

\[
\begin{align*}
\text{Primal} & \quad \min & c_1^T x + \cdots + c_n^T x \\
& \quad \text{s.t.} & A_1 x_1 + \cdots + A_n x_n = b, \\
& & x_i \geq_{Q_{N_i}} 0 (i = 1, \ldots, n); \\
\text{Dual} & \quad \max & b^T y \\
& \quad \text{s.t.} & A_i^T y_i + s_i = c_i (i = 1, \ldots, n), \\
& & s_i \geq_{Q_{N_i}} 0 (i = 1, \ldots, n).
\end{align*}
\]

Here \( N_i \in \mathbb{N} \) is the dimension of variable \( x_i \) and \( s_i (i = 1, \ldots, n) \); \( y \in \mathbb{R}^m \) is a variable for the dual; \( b \in \mathbb{R}^m, c_i \in \mathbb{R}^{N_i}, A_i \in \mathbb{R}^{m \times N_i} (i = 1, \ldots, n) \) are data.

If \( N_i = 2 (i = 1, \ldots, n) \), (1) is a linear program. Write

\[
A \overset{\text{def}}{=} [(a_1)_0, (a_1)_1, \ldots, (a_n)_0, (a_n)_1], \quad c_i \overset{\text{def}}{=} [(c_i)_0, (c_i)_1]^T.
\]

The 2-dimensional second-order cone program is

\[
\begin{align*}
\text{Primal} & \quad \min & \sum_{i=1}^n [(c_i)_0 (x_i)_0 + (c_i)_1 (x_i)_1] \\
& \quad \text{s.t.} & \sum_{i=1}^n [(a_i)_0 (x_i)_0 + (a_i)_1 (x_i)_1] = b, \\
& & (x_i)_0 - (x_i)_1 \geq 0 (i = 1, \ldots, n), \\
& & (x_i)_0 + (x_i)_1 \geq 0 (i = 1, \ldots, n).
\end{align*}
\]

The constraints \((x_i)_0 \geq 0\) are implied in (2), since for each \( i \), it can be obtained by adding \((x_i)_0 - (x_i)_1 \geq 0\) and \((x_i)_0 + (x_i)_1 \geq (i = 1, \ldots, n)\) together. Observe that for any \( i \in \{1, \ldots, n\} \), the number of active constraints determine the states of \( x_i \). If both \((x_i)_0 - (x_i)_1 \geq 0\) and \((x_i)_0 + (x_i)_1 \geq 0\) are active, \( x_i = 0 \); if only one of the constraints is active, \( x_i \neq 0 \) is in the boundary of \( Q_{N_i} \); if none of the constraints is active, \( x_i \) is in the interior of \( Q_{N_i} \). For briefness, we name \((x_i)_0 - (x_i)_1 \geq 0\) as \((-\text{-})\text{-boundary for } x_i \), and \((x_i)_0 + (x_i)_1 \geq 0\) as \((+\text{-})\text{-boundary for } x_i \).

The dual to (2) is:

\[
\begin{align*}
\text{Dual} & \quad \max & b^T y \\
& \quad \text{s.t.} & (a_i)_0^T y + (s_i)_0 = (c_i)_0 (i = 1, \ldots, n), \\
& & (a_i)_1^T y + (s_i)_1 = (c_i)_1 (i = 1, \ldots, n), \\
& & (s_i)_0 - (s_i)_1 \geq 0 (i = 1, \ldots, n), \\
& & (s_i)_0 + (s_i)_1 \geq 0 (i = 1, \ldots, n).
\end{align*}
\]

Since (2) and (3) are not in standard LP form, next we will transform them into the standard LP form and give some properties of the transformation.

2.2 The Variable Transformation

Set

\[
K \overset{\text{def}}{=} \begin{pmatrix}
\frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \\
-\frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2}
\end{pmatrix}.
\]

Then \( K \) is an orthogonal matrix rotating the axes 45° clockwise.

**Properties of \( K \).**

For \( i = 1, \ldots, n \), let \( v_i = K x_i \). Then:

\[
\begin{align*}
\sum_{i=1}^n (v_i)_0 &= \sum_{i=1}^n (a_i)_0 (x_i)_0 + (s_i)_0 = \sum_{i=1}^n (a_i)_0 (x_i)_0 + \sum_{i=1}^n (s_i)_0 \quad (i = 1, \ldots, n), \\
\sum_{i=1}^n (v_i)_1 &= \sum_{i=1}^n (a_i)_1 (x_i)_0 + (s_i)_1 = \sum_{i=1}^n (a_i)_1 (x_i)_0 + \sum_{i=1}^n (s_i)_1 \quad (i = 1, \ldots, n).
\end{align*}
\]
1. $|(v_i)_0|$ is the distance of $x_i$ from the boundary $(x_i)_0 = (x_i)_1$; $|(v_i)_1|$ is the distance of $x_i$ from the boundary $(x_i)_0 = -(x_i)_1$.

2. $\sqrt{2}v$ are the two eigenvalues (see [2]) of $x$.

3. $x_i \geq 0$ if $v_i \geq 0$.

Remark 2.1 Let

$$\hat{K} \overset{\text{def}}{=} \begin{pmatrix} \sqrt{2}/2 & \sqrt{2}/2 \\ \sqrt{2}/2 & -\sqrt{2}/2 \end{pmatrix},$$

i.e., $\sqrt{2}\hat{K}$ is a Hadamard matrix of order 2. We can also use the transformation $\hat{K}$; and the analysis below can be adjusted to $\hat{K}$ as well.

Let $P$ be a block diagonal matrix with each block $K$. Denote $v \overset{\text{def}}{=} (v_1; \ldots; v_n)$. Then $v = Px$.

Let $c \overset{\text{def}}{=} Pc$, $\tilde{A} \overset{\text{def}}{=} APT$. Then (2)-(3) can be transformed into the following standard linear program:

\begin{align*}
\text{Primal} & \quad \min \quad \tilde{c}^T v \\
& \text{s.t.} \quad \tilde{A}v = b, \\
& \quad v \geq 0;
\end{align*}

\begin{align*}
\text{Dual} & \quad \max \quad b^T y \\
& \text{s.t.} \quad \tilde{A}^T y + w = \tilde{c}, \\
& \quad w \geq 0.
\end{align*}

The solutions to (2)-(3) and that to (4) are closely related, as is stated below.

\textbf{Relations between the solutions to LP and SOCP.}

1. A solution to (4) and a solution to (2)-(3) is related by $(v, w, y) = (Px, Ps, y)$. Under this relation, (4) and (2)-(3) have the same objective value.

2. As $v_i$ does to $x_i$, $w_i$ measures the feasibility of $s_i$ with regard to the second-order cone constraints.

3. $V_i w_i = 0 \iff x_i \circ s_i = 0$.

Here $v_i \cdot w_i = \begin{pmatrix} (v_i)_0 (w_i)_0 \\ (v_i)_1 (w_i)_1 \end{pmatrix}$, $x_i \circ s_i = \begin{pmatrix} (x_i)_0 (s_i)_0 + (x_i)_1 (s_i)_1 \\ (x_i)_1 (s_i)_0 + (x_i)_0 (s_i)_1 \end{pmatrix}$.

4. Complementarity for $(v, w)$ in LP is satisfied iff complementarity for $(x, s)$ in SOCP is satisfied.

This can be seen by the above relation 3 and the property 3 for $K$.

\textbf{Strong duality}

The above transformation implies that the strong duality for the 2-dimensional SOCP holds if (i) both the primal (2) and the dual (3) have feasible solutions, or (ii) the primal (2) 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 [2].

### 3 The Simplex Method

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

Without loss of generality, we assume $A$ has full row rank, since otherwise either the linear constraints are inconsistent or some linear constraints are redundant. Then there is an $m \times m$ submatrix $A_B$ of $A$ that is nonsingular. Since $P$ is nonsingular, $A_B$ is nonsingular, too.
In a 2-dimensional SOCP, a zero variable means both (−) and (+) constraints are active; a boundary variable has only one active constraint; an interior variable has no active constraint. For a solution \( x \), let \( B_x \) represent the number of its boundary blocks, \( I_x \) the number of its interior blocks, and \( O_x \) the number of its zero blocks. If a linear program is solvable, then it has a basic optimal solution. So if a 2-dimensional second order cone program is solvable, it has a basic optimal solution \( x^* \) in the sense that

\[
B_{x^*} + 2I_{x^*} \leq m,
\]

and \( A_B \) is an \( m \times m \) submatrix of \( A \) that is nonsingular, where the indices of \( A_B \) are those for the constraints corresponding to the zero blocks and those for the inactive constraints of the boundary blocks. Next we interpret the primal simplex method with Bland’s pivoting rule. Other simplex methods can be explained in a similar way.

**The primal simplex method with Bland’s pivoting rule**

1. Solve a phase I problem to get a basic feasible solution to the primal of (4). Assume the corresponding partition of index set \( I \) \( \{i(k) \mid i \in \{1, \ldots, n\}, k \in \{-, +\}\} \) \( i \) is the block index, \( k \) indicates the boundary constraints (+) or (−)) is B and N, where \( A_B \in \mathbb{R}^{m \times m} \) is nonsingular. Let \( v_B \) and \( v_N \) be the basic and nonbasic variables, i.e. \( v_B = A_B^{-1}b - A_B^{-1}A_N v_N \).

2. If \( \tilde{c}_N - (A_B^{-1}A_N)^T \tilde{c}_B \geq 0 \), \( P^Tv \) is optimal for (2). Stop. Otherwise, there exists index \( i(k) \) such that \( (\tilde{c}_N - (A_B^{-1}A_N)^T \tilde{c}_B)_{i(k)} \) : that indicates if \( x_i \) is moved away from the boundary \( k \), the objective may be decreased.

3. Check the columns of \( (A_B^{-1}A_N)_{i(k)} \) for such \( i(k) \)’s. If there exists an \( i(k) \) such that \( (A_B^{-1}A_N)_{i(k)} \leq 0 \), then the problem is unbounded, i.e. \( x_i \) can be moved arbitrarily away from the boundary \( k \) to decrease the objective infinitely. Otherwise, from the \( i(k) \)’s choose the smallest index \( \bar{i}(k) \); from the indices \( j(l) \)’s with \( (A_B^{-1}A_N)_{j(l), i(k)} > 0 \) choose the smallest index \( \bar{j}(l) \). Move \( x_j \) to boundary \( \bar{l} \), and move \( x_i \) away from boundary \( \bar{k} \) at a distance \( (A_B^{-1}b)_{j(l)} / (A_B^{-1}A_N)_{j(l), i(k)} \).

4. Go to step 2 with the new basic, nonbasic variables and coefficient matrix.

In the above algorithm, each step 2 affects only two constraints: one active constraint will be inactive, and one inactive constraints will be active. Next, we will consider how the status of \( x \) is affected by the pivoting.

**The status of the variable**

1. \( \bar{i} = \bar{j} \)
   
   In this case, \( x_i \) must be moved from one side of the boundary to the other side of the boundary of \( Q \).

2. \( \bar{i} \neq \bar{j} \)
   
   That means the pivot affects two variables. Since the total number of active constraints for \( x_i \) and \( x_j \) is unchanged after pivoting, and this number can be only 1, 2, 3.
   
   (a) **The number is 1**: This means that the pivoting makes an interior variable boundary and a boundary variable interior.
   
   (b) **The number is 2**: This means that after the pivoting, a 0 and an interior variable become two boundary variables, or vice versa.
   
   (c) **The number is 3**: This means that a 0 variable is changed to a boundary variable, and a boundary variable is changed to a zero variable by the pivoting.
Other methods for linear program, such as dual simplex algorithm, primal-dual simplex algorithm (see [8]) can also be applied to the 2-dimensional second-order cone program.

**Sensitive analysis**

We can also perform sensitive and parameter analysis on the 2-dimensional second-order cone programming. Given a basic optimal solution for the primal-dual pair (2)-(3), the corresponding \((v^*, w^*)\) is the following.

\[
\begin{align*}
v_B^* &= \hat{A}^{-1}_B b, \\
w_N^* &= (\hat{A}^{-1}_B)^T \hat{c}_B - \hat{c}_N, \\
\zeta^* &= \hat{c}^T \hat{A}^{-1}_B b,
\end{align*}
\]

where \(\zeta^*\) is the objective value. Assume \(b\) is changed to \(b + t \Delta b\), if

\[
\left( \min_{j(l) \in B} - \frac{\Delta b_{j(l)}}{(\hat{A}^{-1}_B b)_{j(l)}} \right)^{-1} \leq t \leq \left( \max_{j(l) \in B} - \frac{(\hat{A}^{-1}_B b)_{j(l)}}{(\hat{A}^{-1}_B b)_{j(l)}} \right)^{-1},
\]

the states of \(x\) will remain unchanged, so will those of \(s\). The objective will be \(\zeta^* + t \hat{c}^T \hat{A}^{-1}_B b\).

Assume \(c\) is changed to \(c + t \Delta c\), set \(\Delta w \stackrel{\text{def}}{=} \left( \hat{A}^{-1}_B \hat{A}_N \right)^T \Delta c_B - \Delta c_N\), if

\[
\left( \min_{i(k) \in N} - \frac{\Delta w_{i(k)}}{w_{i(k)}} \right)^{-1} \leq t \leq \left( \max_{i(k) \in N} - \frac{\Delta w_{i(k)}}{w_{i(k)}} \right)^{-1},
\]

the states of \(s\) will remain unchanged, so will those of \(x\). The objective will be \(\zeta^* + t \hat{c}^T \hat{A}^{-1}_B b\).

### 4 Application

Below are some applications of the 2-dimensional second-order cone programming.

The supreme of the absolute value \(|v|\) can be formulated as a 2-dimensional SOCP problem.

\[
\begin{align*}
\min_{v_0} & \quad v_0 \\
\text{s.t.} & \quad (v_0, v)^T \succeq 0.
\end{align*}
\]

Let \(\psi_i = A_i x + b_i \in \mathbb{R}^{n_i}, i = 1, \ldots, k\). Then the following norm minimization problem can be cast as SOCP problems.

**Minimize the Sum of Norms.**

1. **Minimize the sum of \(L^1\)-norms.**

The problem \(\min \|\psi_1\|_1 + \cdots + \|\psi_k\|_1\) can be formulated as follows.

\[
\begin{align*}
\min & \quad \sum_{i=1}^k \sum_{j=1}^{n_i} (x_{ij})_0 \\
\text{s.t.} & \quad A_i x + b_i = \psi_i \quad (i = 1, \ldots, k) \\
& \quad [(x_{ij})_0, (\tilde{e}_j)_j]^T \succeq 0 \quad (i = 1, \ldots, k; j = 1, \ldots, n_i).
\end{align*}
\]

2. **Minimize the sum of \(L^\infty\)-norms.**

As that above, \(\min \|\psi_1\|_\infty + \cdots + \|\psi_k\|_\infty\) can be cast as a 2-dimensional SOCP problem.

\[
\begin{align*}
\min & \quad \sum_{i=1}^k (x_{ij})_0 \\
\text{s.t.} & \quad A_i x + b_i = \psi_i \quad (i = 1, \ldots, k) \\
& \quad [(x_{ij})_0, (\tilde{e}_j)_j]^T \succeq 0 \quad (i = 1, \ldots, k; j = 1, \ldots, n_i).
\end{align*}
\]

**Minimize the Largest Norm.**
1. Minimize the largest $L_1$-norm.
The problem $\min \max_{1 \leq i \leq k} \| \bar{v}_i \|_1$ can be written as the follow.

$$
\min \ t \\
\text{s.t.} \ A_i v + b_i = \bar{v}_i \quad (i = 1, \ldots, k),
\quad [\langle x_{ij} \rangle_0, (\bar{v}_i)_{ij}]^T \geq \mathbf{0} \quad (i = 1, \ldots, k; \ j = 1, \ldots, n_i),
\quad [t, \sum_{j=1}^{n_i} (x_{ij})_0]^T \geq \mathbf{0} \quad (i = 1, \ldots, k).
$$

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

$$
\min \ t \\
\text{s.t.} \ A_i v + b_i = \bar{v}_i \quad (i = 1, \ldots, k),
\quad [t, (\bar{v}_i)_{ij}]^T \geq \mathbf{0} \quad (i = 1, \ldots, k; \ j = 1, \ldots, n_i).
$$

Minimize the Sum of $r$ largest norms.

1. Minimize the sum of $r$ largest $L_1$-norm.
The problem $\min \sum_{i=1}^{r} \| \bar{v}_i \|_1$ can be cast as the following.

$$
\min \ \sum_{i=1}^{r} u_i + rt \\
\text{s.t.} \ A_i x + b_i = \bar{v}_i \quad (i = 1, \ldots, k),
\quad [\langle x_{ij} \rangle_0, (\bar{v}_i)_{ij}]^T \geq \mathbf{0} \quad (i = 1, \ldots, k; \ j = 1, \ldots, n_j),
\quad [t + u_i, \sum_{j=1}^{n_i} (x_{ij})_0]^T \geq \mathbf{0} \quad (i = 1, \ldots, k).
$$

2. Minimize the sum of $r$ largest $L_\infty$-norm.
The problem $\min \sum_{i=1}^{r} \| \bar{v}_i \|_\infty$ can be formulated as the following.

$$
\min \ \sum_{i=1}^{r} u_i + rt \\
\text{s.t.} \ A_i x + b_i = \bar{v}_i \quad (i = 1, \ldots, k),
\quad [\langle x_{ij} \rangle_0, (\bar{v}_i)_{ij}]^T \geq \mathbf{0} \quad (i = 1, \ldots, k; \ j = 1, \ldots, n_j),
\quad [t + u_i, (x_{ij})_0]^T \geq \mathbf{0} \quad (i = 1, \ldots, k).
$$

References

[1] I. Adler and F. Alizadeh. Primal-dual interior point algorithms for convex quadratically con-
strained and semidefinite optimization problems. Technical Report RRR 46-95, RUTCOR, Rut-
gers University, 1995.


