Liang Fang; Guoping He; Li Sun

A globally convergent non-interior point algorithm with full Newton step for second-order cone programming

Applications of Mathematics, Vol. 54 (2009), No. 5, 447--464

Persistent URL: http://dml.cz/dmlcz/140378

54 (2009) APPLICATIONS OF MATHEMATICS No. 5, 447–464

A GLOBALLY CONVERGENT NON-INTERIOR POINT ALGORITHM WITH FULL NEWTON STEP FOR SECOND-ORDER CONE PROGRAMMING*

Liang Fang, Tai'an and Shanghai, Guoping He, Qingdao and Shanghai,
Li Sun, Tai'an

(Received June 10, 2008)

Abstract. A non-interior point algorithm based on projection for second-order cone programming problems is proposed and analyzed. The main idea of the algorithm is that we cast the complementary equation in the primal-dual optimality conditions as a projection equation. By using this reformulation, we only need to solve a system of linear equations with the same coefficient matrix and compute two simple projections at each iteration, without performing any line search. This algorithm can start from an arbitrary point, and does not require the row vectors of A to be linearly independent. We prove that our algorithm is globally convergent under weak conditions. Preliminary numerical results demonstrate the effectiveness of our algorithm.

Keywords: non-interior point algorithm, second-order cone programming, Jordan product, optimality condition, central path

MSC 2010: 90C25, 90C30, 90C51, 65K05, 65Y20

1. Introduction

A second-order cone programming (SOCP) is to minimize a linear function over the intersection of an affine space with the Cartesian product of a finite number of second-order cones. The SOCPs have wide range of applications in many fields, such as engineering, control, finance, robust optimization and combinatorial optimization (see [1], [14], [15], [17], [20], [32], [33]).

* This work was supported by National Natural Science Foundation of China (10571109) and Natural Science Foundation of Shandong (Y2008A01).

Throughout this paper, we consider the SOCP problem with a single second-order cone

$$\text{(1)} \qquad \text{(P)} \qquad \min\{\langle c, x\rangle\colon\ Ax = b,\ x \in \mathcal{Q}\},$$

where $A \in \mathbb{R}^{m\times n}$, $c \in \mathbb{R}^n$, $b \in \mathbb{R}^m$ are given data, $\langle\cdot,\cdot\rangle$ is the Euclidean inner product, and $\mathcal{Q}$ is a second-order cone (SOC) with dimension n which is defined by

$$\mathcal{Q} = \{(x_0; \bar{x}) \in \mathbb{R} \times \mathbb{R}^{n-1}\colon\ x_0 \geqslant \|\bar{x}\|\},$$

where x_0 is the first element of x, $\bar{x}$ is the vector containing the remaining elements of x, $\|\cdot\|$ refers to the standard Euclidean norm. For simplicity, we use "," to join vectors and matrices in a row and ";" to join them in a column. Thus, for instance, for vectors x, y, and z we use $(x; y; z)$ to represent $(x^T, y^T, z^T)^T$.

It is well known that SOC $\mathcal{Q} \subseteq \mathbb{R}^n$ is a closed, pointed (i.e., $\mathcal{Q} \cap (-\mathcal{Q}) = \{0\}$) and convex cone. Hence, SOCP problems are convex optimization problems and the SOC $\mathcal{Q}$ is self-dual, that is,

$$\mathcal{Q} = \mathcal{Q}^* = \{s \in \mathbb{R}^n\colon\ s^T x \geqslant 0,\ \text{for each } x \in \mathcal{Q}\}.$$

Thus the dual problem of (P) is

$$\text{(2)} \qquad \text{(D)} \qquad \max\{b^T y\colon\ A^T y + s = c,\ s \in \mathcal{Q}\},$$

where $y \in \mathbb{R}^m$ and $s \in \mathcal{Q}$ is slack variable. The symbol $(\mathbb{R}^n, \circ)$ stands for a Euclidean Jordan algebra, where "$\circ$" is a bilinear mapping named the Jordan product and defined by

$$\text{(3)} \qquad x \circ s = (x^T s; x_0\bar{s} + s_0\bar{x})$$

for any $x = (x_0; \bar{x})$ and $s = (s_0; \bar{s}) \in \mathbb{R} \times \mathbb{R}^{n-1}$. The vector $e_n = (1; 0) \in \mathbb{R} \times \mathbb{R}^{n-1}$ is the identity element of $(\mathbb{R}^n, \circ)$. Each vector is indexed from 0. We use lower case letters such as x, s for column vectors, and upper case letters A, B for matrices, $\mathbf{0}_n$ denotes the square matrix whose dimension is n and elements are all zeros, and 0 denotes the vector of all zeros with suitable dimension. Subscripted vectors such as x_i represent the ith element of x. We use I for the identity matrices; in all cases the dimensions of vectors and matrices can be discerned from the context. If $\mathcal{K} \subseteq \mathbb{R}^k$ and $\mathcal{L} \subseteq \mathbb{R}^l$, then $\mathcal{K} \times \mathcal{L} = \{(x; y)\colon\ x \in \mathcal{K},\ y \in \mathcal{L}\}$ is the Cartesian product of $\mathcal{K}$ and $\mathcal{L}$. Let $\operatorname{bd}\mathcal{Q} = \{x \in \mathcal{Q}\colon\ x_0 = \|\bar{x}\| \text{ and } x \neq 0\}$ denote the boundary of $\mathcal{Q}$, while the interior of $\mathcal{Q}$ is denoted by $\operatorname{int}\mathcal{Q} = \{x \in \mathcal{Q}\colon\ x_0 > \|\bar{x}\|\}$. For $A \in \mathbb{R}^{n\times n}$, $A \succcurlyeq \mathbf{0}$

$(A \succ \mathbf{0})$ means A is positive semidefinite (positive definite). It is well known that $\mathcal{Q}$ induces a partial ordering on $\mathbb{R}^n$:

$$x \succcurlyeq_{\mathcal{Q}} y \ \text{ iff } \ x - y \in \mathcal{Q}, \quad \text{and} \quad x \succ_{\mathcal{Q}} y \ \text{ iff } \ x - y \in \operatorname{int} \mathcal{Q}.$$

The relations "$\preccurlyeq_{\mathcal{Q}}$" and "$\prec_{\mathcal{Q}}$" are defined similarly. In analogy to matrices, we call $x \in \mathcal{Q}$ (i.e., $x \succcurlyeq 0$) positive semidefinite and $x \in \operatorname{int}\mathcal{Q}$ (i.e., $x \succ 0$) positive definite [1]. For the sake of convenience, we denote $w^k := (x^k, y^k, s^k)$, where k denotes the iteration index, and let

$$\begin{aligned}
&\mathcal{F}(\mathrm{P}) = \{x \in \mathbb{R}^n \colon \ Ax = b, \ x \in \mathcal{Q}\}, \\
&\mathcal{F}(\mathrm{D}) = \{(y, s) \in \mathbb{R}^m \times \mathbb{R}^n \colon \ A^T y + s = c, \ s \in \mathcal{Q}\}
\end{aligned}$$

represent the feasible sets of the (P) and (D), respectively. At the same time, the interior feasible solutions of (P) and (D) are represented by

$$\begin{aligned}
&\mathcal{F}^0(\mathrm{P}) = \{x \in \mathbb{R}^n \colon \ Ax = b, \ x \in \operatorname{int} \mathcal{Q}\}, \\
&\mathcal{F}^0(\mathrm{D}) = \{(y, s) \in \mathbb{R}^m \times \mathbb{R}^n \colon \ A^T y + s = c, \ s \in \operatorname{int} \mathcal{Q}\}.
\end{aligned}$$

Note that our analysis can be easily extended to the general case in the Cartesian product of a finite number of SOCs.

For any $x \in \mathbb{R}^n$, we have the basic conclusion:

$$x \in \mathcal{Q} \Leftrightarrow \langle x, s\rangle \geqslant 0 \quad \forall\, s \in \mathcal{Q}. \tag{4}$$

Associated with each vector $x \in \mathbb{R}^n$, there is an *arrow-shaped matrix* defined as

$$\operatorname{Arw}(x) = \begin{pmatrix} x_0 & \bar{x}^T \\ \bar{x} & x_0 I \end{pmatrix}.$$

It is easy to see that $\operatorname{Arw}(x) \succcurlyeq 0$ if and only if either $x = 0$, or $x_0 > 0$ and the Schur complement $x_0 - \bar{x}^T (x_0 I)^{-1} \bar{x} \geqslant 0$, which implies that $x \succcurlyeq_{\mathcal{Q}} 0$ ($x \succ_{\mathcal{Q}} 0$) if and only if $\operatorname{Arw}(x) \succcurlyeq 0$ ($\operatorname{Arw}(x) \succ 0$). As a consequence, we conclude that SOCP is a special case of semidefinite programming. However, the algorithm which is reliable for solving SOCP warrants our attention.

For a given primal-dual feasible point (x, y, s), $\langle x, s\rangle$ is called the duality gap due to the famous weak dual theorem, i.e., $\langle x, s\rangle \geqslant 0$, which implies that

$$\langle c, x\rangle - \langle b, y\rangle = \langle A^T y + s, x\rangle - \langle Ax, y\rangle = \langle x, s\rangle \geqslant 0.$$

Note that $\langle x, s\rangle = 0$ is sufficient for optimality for a feasible point (x, y, s).

It is well known that under a suitable condition, such as the *Slater constraint qualification*, the SOCP is equivalent to its *optimality conditions*

$$
\begin{aligned}
Ax &= b,\\
A^T y + s &= c,\\
\langle x, s\rangle = 0, \quad x, s &\in \mathcal{Q}, \quad y \in \mathbb{R}^m,
\end{aligned} \tag{5}
$$

where $\langle x, s\rangle = 0$ is usually referred to as the *complementarity condition*.

Over the past decades, the primal-dual interior point method (IPM) has been developed for all kinds of nonlinear optimization problems including SOCPs (see, e.g., [1], [3], [5]–[8], [14]–[15], [17]–[20], [22]–[25], [28]–[30], [32]–[34] and references therein). Numerous applications have been discussed in [17], [20], [27], [32]. Many researchers indicated that the primal-dual IPM had the highly theoretical efficiency for SOCPs.

Recently, motivated by the smoothing-type methods for linear programming and complementarity problems, many methods for solving SOCPs have been proposed in [2], [4], [9], [11], [13], [16], [26]. They include reformulating SOC constraints as smooth convex constraints, smoothing Newton methods, and smoothing-regularization methods. These methods require solving a nontrivial system of linear equations at each iteration. The main idea of these methods is that the optimality conditions or central path conditions are reformulated as a nonlinear equation, which excludes the inequality constraints such as $x \succcurlyeq_{\mathcal{Q}} 0$, $s \succcurlyeq_{\mathcal{Q}} 0$ or $x \succ_{\mathcal{Q}} 0$, $s \succ_{\mathcal{Q}} 0$.

Under mild assumptions, both IPM and the smoothing method are globally convergent. It should be noted that both the methods require the linear independence of the row vectors of the matrix A, and a suitable step length obtained by the line search. Moreover, a feasible starting point is needed except infeasible IPMs (see [21], [22]).

The non-interior point algorithm to be discussed here is motivated by the alternating direction methods for variational inequality problems (see [5], [10], [12], [14], [17], [20], [22], [31]). The new algorithm is based on the optimality conditions (5) and the main difference from IPMs and smoothing methods is that we reformulate the complementarity condition as a projection equation. It is shown that our algorithm has the following good properties:

(i) The algorithm can start from an arbitrary point.

(ii) The algorithm needs to solve only one linear system of equations and compute two simple projections at each iteration.

(iii) It does not require any line search, i.e., full Newton step is taken at each iteration.

(iv) It does not require A to be of full row rank.

(v) The generated sequence converges to the accumulation point globally without strict complementarity, which is stronger than the corresponding results for IPMs and smoothing-type methods.

The paper is organized as follows. In Section 2, we give the equivalent formulation of the optimality conditions. A new algorithm is stated in Section 3. In Section 4, we analyze the global convergence of our algorithm. Preliminary numerical tests are shown in Section 5. Finally, some conclusions and suggestions for future research are summarized in the last section.

2. Equivalent formulation of optimality conditions

In this section we give the equivalent formulation of the optimality conditions in (5). To this end, we exploit the following result.

Proposition 2.1. *Let* $x \succcurlyeq_Q 0$, $s \succcurlyeq_Q 0$, *then* $\langle x, s\rangle = 0$ *is equivalent to* $x \circ s = 0$.

Proof. If $x_0 = 0$ or $s_0 = 0$, then the conclusion is obvious. Now, we assume that $x_0 > 0$ and $s_0 > 0$. From $x \succcurlyeq_Q 0$, $s \succcurlyeq_Q 0$, we have

$$x_0^2 \geqslant \|\bar{x}\|^2 = \sum_{i=1}^{n-1} x_i^2, \tag{6}$$

$$s_0^2 \geqslant \|\bar{s}\|^2 = \sum_{i=1}^{n-1} s_i^2, \tag{7}$$

which yields

$$x_0^2 \geqslant x_0^2 \sum_{i=1}^{n-1} \frac{s_i^2}{s_0^2}. \tag{8}$$

By $\langle x, s\rangle = 0$ we have $-x_0 s_0 = x_1 s_1 + x_2 s_2 + \ldots + x_{n-1} s_{n-1}$ which is equivalent to

$$-2x_0^2 = \sum_{i=1}^{n-1} \frac{2x_0 x_i s_i}{s_0}. \tag{9}$$

Adding (6), (8), and (9) together gives

$$\sum_{i=1}^{n-1} \Big(x_i + \frac{x_0 s_i}{s_0}\Big)^2 \leqslant 0,$$

from which we obtain $x_0 s_i + s_0 x_i = 0$, $i = 1, \ldots, n-1$. Therefore, we have $\langle x, s\rangle = 0$ if and only if $x \circ s = 0$. □

Throughout the paper, we make the following assumption:

Assumption 2.1. $\mathcal{F}^0(\mathrm{P}) \times \mathcal{F}^0(\mathrm{D}) \neq \emptyset$.

Note that under Assumption 2.1, strong dual theorem holds, i.e., both (P) and (D) have optimal solutions and their optimal values are coincident. Therefore, the optimality condition (5) is equivalent to

$$
\begin{aligned}
Ax &= b, \\
A^T y + s &= c, \\
x \circ s = 0, \quad x, s &\in \mathcal{Q}, \quad y \in \mathbb{R}^m.
\end{aligned}
\tag{10}
$$

Several researchers suggest solving the optimality conditions (5) by IPMs motivated by the groundbreaking work of Nesterov and Nemirovskii (see [1], [15] and references therein). They typically consider the following perturbed optimality conditions, which are usually called the *central path conditions*:

$$
\begin{aligned}
Ax &= b, \\
A^T y + s &= c, \\
x \circ s = \mu e_n, \quad x, s &\in \operatorname{int} \mathcal{Q}, \quad y \in \mathbb{R}^m,
\end{aligned}
\tag{11}
$$

where $\mu > 0$ is a parameter. IPMs usually apply a Newton-type method to the central path conditions and then deal with $x \succ 0$ and $s \succ 0$ explicitly by a suitable line search.

Associated with SOC $\mathcal{Q}$, we define the *spectral decomposition* of any vector $x = (x_0; \bar{x}) \in \mathbb{R} \times \mathbb{R}^{n-1}$ as

$$
x = \lambda_1 u_1 + \lambda_2 u_2, \tag{12}
$$

where the spectral values λ_1, λ_2 and the associated spectral vectors u_1, u_2 of x are given by

$$
\lambda_i = x_0 + (-1)^{i+1} \|\bar{x}\|, \tag{13}
$$

$$
u_i = \begin{cases} \dfrac{1}{2}\Big(1; (-1)^{i+1} \dfrac{\bar{x}}{\|\bar{x}\|}\Big), & \bar{x} \neq 0; \\[2ex] \dfrac{1}{2}(1; (-1)^{i+1} \omega), & \bar{x} = 0, \end{cases} \tag{14}
$$

for $i = 1, 2$, with any $\omega \in \mathbb{R}^{n-1}$ such that $\|\omega\| = 1$.

Since $\mathcal{Q}$ is an SOC, it is a nonempty closed convex set in $\mathbb{R}^n$. The *orthogonal projection* from $\mathbb{R}^n$ into $\mathcal{Q}$ is defined by

$$P_{\mathcal{Q}}(x) = \arg\min\{\|\omega - x\|\colon\ \omega \in \mathcal{Q}\}, \quad \forall\, x \in \mathbb{R}^n. \tag{15}$$

A basic property of the projection mapping is shown in Lemma 2.1.

Lemma 2.1. *For any* $u, v \in \mathbb{R}^n$, $w \in \mathcal{Q}$, *we have*

$$\langle v - P_{\mathcal{Q}}(v), P_{\mathcal{Q}}(v) - w\rangle \geqslant 0, \tag{16}$$

$$\|P_{\mathcal{Q}}(u) - P_{\mathcal{Q}}(v)\| \leqslant \|u - v\|. \tag{17}$$

Moreover, $P_{\mathcal{Q}}(\cdot)$ has the following important properties.

Proposition 2.2. *For any* $x = \lambda_1 u_1 + \lambda_2 u_2 \in \mathbb{R}^n$, λ_i *and* u_i *being defined by* (13) *and* (14), *we have*

$$P_{\mathcal{Q}}(x) = \lambda_1^+ u_1 + \lambda_2^+ u_2, \tag{18}$$

where $\lambda_i^+ = \max\{\lambda_i, 0\}$, $i = 1, 2$.

Proof. It is obvious that the pair of vectors $\{u_1, u_2\}$ is a *Jordan frame.* The superplane

$$\mathcal{S} = \{z = \nu_1 u_1 + \nu_2 u_2\colon\ \nu_1, \nu_2 \in \mathbb{R}\} \tag{19}$$

is a complete normed linear space in $\mathbb{R}^n$.

For any $\gamma \in \mathcal{Q}$, it follows from the orthogonal decomposition theorem that γ can be decomposed into $\gamma = \gamma_1 + \gamma_2$, where $\gamma_1 = P_{\mathcal{S}}(\gamma) \in \mathcal{S} \subseteq \mathcal{Q}$, $\gamma_2 \in \mathcal{S}^\perp$, i.e., γ_2 is in the orthogonal complement of $\mathcal{S}$.

Assume $\gamma_1 = \tilde\lambda_1 u_1 + \tilde\lambda_2 u_2$, $\tilde\lambda_1, \tilde\lambda_2 \geqslant 0$. Then we have

$$\begin{aligned}\|x - \gamma\|^2 &= \|(\lambda_1 - \tilde\lambda_1)u_1 + (\lambda_2 - \tilde\lambda_2)u_2 - \gamma_2\|^2\\ &= \langle(\lambda_1 - \tilde\lambda_1)u_1 + (\lambda_2 - \tilde\lambda_2)u_2 - \gamma_2, (\lambda_1 - \tilde\lambda_1)u_1 + (\lambda_2 - \tilde\lambda_2)u_2 - \gamma_2\rangle\\ &= \frac{1}{2}(\lambda_1 - \tilde\lambda_1)^2 + \frac{1}{2}(\lambda_2 - \tilde\lambda_2)^2 + \|\gamma_2\|^2,\end{aligned}$$

where the third equality follows from the fact that $\langle u_i, u_i\rangle = 1/2$, $\langle u_1, u_2\rangle = 0$, and $\langle u_i, \gamma_2\rangle = 0$, $i = 1, 2$. The right-hand side is minimized by the γ such that $\tilde\lambda_i = \max\{\lambda_i, 0\} = \lambda_i^+$, $i = 1, 2$, and $\gamma_2 = 0$. Thus the proof is completed. □

Note that Proposition 2.2 offers a simple way to compute the projection $P_{\mathcal{Q}}(x)$ via the spectral decomposition of x.

Remark 2.1. It is easy to conclude that

(1) if $x \in \mathcal{Q}$, then $P_{\mathcal{Q}}(x) = x$;

(2) if $x \in -\mathcal{Q}^*$, then $P_{\mathcal{Q}}(x) = 0$;

(3) if $x \in \mathbb{R}^n - (\mathcal{Q} \cup (-\mathcal{Q}^*))$, then $P_{\mathcal{Q}}(x) = \lambda_i u_i$, where λ_i is the only positive characteristic eigenvalue (the other one is negative), whose characteristic vector is u_i, $1 \leqslant i \leqslant 2$. Therefore, the projection of $x = (x_0; \bar{x}) \in \mathbb{R} \times \mathbb{R}^{n-1}$ on SOC $\mathcal{Q}$ can be expressed by

$$P_{\mathcal{Q}}(x) = \begin{cases} x, & \lambda_1 \geqslant 0,\ \lambda_2 \geqslant 0; \\ \dfrac{1}{2}(x_0 + \|\bar{x}\|)\Big(1; \dfrac{\bar{x}}{\|\bar{x}\|}\Big), & \lambda_1 > 0,\ \lambda_2 < 0; \\ 0, & \lambda_1 \leqslant 0,\ \lambda_2 \leqslant 0, \end{cases} \tag{20}$$

where λ_1, λ_2 are the spectral values of x defined by (13).

The following theorem plays an important role in our reformulation.

Theorem 2.1. *For any $(x, s) \in \mathbb{R}^n \times \mathbb{R}^n$, we have*

$$x, s \in \mathcal{Q}, \quad \langle x, s \rangle = 0 \Leftrightarrow s = P_{\mathcal{Q}}(s - x). \tag{21}$$

Proof. First consider the "only if" part. If $(x, s) \in \mathcal{Q} \times \mathcal{Q}$ satisfies $\langle x, s \rangle = 0$, then for any $c \succcurlyeq_{\mathcal{Q}} 0$ we have

$$\|(s - x) - c\|^2 = \|s - c\|^2 + 2\langle c - s, x \rangle + \|x\|^2 = \|s - c\|^2 + 2\langle c, x \rangle + \|x\|^2. \tag{22}$$

Since $x, c \succcurlyeq_{\mathcal{Q}} 0$, by (4) we have $\langle c, x \rangle \geqslant 0$. The right-hand side attains its minimum $\|x\|^2$ at $c = s$, which means $s = P_{\mathcal{Q}}(s - x)$.

Next, we consider the "if" part. If $(x, s) \in \mathbb{R}^n \times \mathbb{R}^n$ satisfies $s = P_{\mathcal{Q}}(s - x)$, then $s \succcurlyeq_{\mathcal{Q}} 0$ and

$$0 \leqslant \|(s - x) - c\|^2 - \|x\|^2 = \|s - c\|^2 + 2\langle c - s, x \rangle \quad \forall\, c \succcurlyeq_{\mathcal{Q}} 0. \tag{23}$$

For any $\omega \succcurlyeq_{\mathcal{Q}} 0$ and any $t \in (0, +\infty)$, we have $c = s + t\omega \succcurlyeq_{\mathcal{Q}} 0$. The equation (23) yields

$$t^2\|\omega\|^2 + 2t\langle \omega, x \rangle \geqslant 0.$$

Dividing both sides by t and letting $t \to 0$ yields $\langle \omega, x \rangle \geqslant 0$ for all $\omega \succcurlyeq_{\mathcal{Q}} 0$. Also, by (4) we have $x \succcurlyeq_{\mathcal{Q}} 0$. Similarly, for any $t \in (0, 1]$ we have

$$c = (1 - t)s \succcurlyeq_{\mathcal{Q}} 0.$$

It follows from (23) that

$$t^2\|s\|^2 - 2t\langle s, x\rangle \geqslant 0.$$

Dividing both sides by t and letting $t \to 0$ yields $-\langle s, x\rangle \geqslant 0$. Since $x, s \in \mathcal{Q}$, by (4) we get $\langle x, s\rangle \geqslant 0$, which implies $\langle x, s\rangle = 0$. The proof is completed. □

From Theorem 2.1 we obtain the following equivalent reformulation of the optimal conditions (5):

$$\Phi(w) := \Phi(x, y, s) = \begin{pmatrix} Ax - b \\ c - A^T y - s \\ s - P_{\mathcal{Q}}(s - x) \end{pmatrix} = 0, \tag{24}$$

where $x, s \in \mathcal{Q}$, $y \in \mathbb{R}^m$.

3. Description of the algorithm

The aim of this section is to propose the non-interior point method for solving SOCPs. From (24) we can see that for a given element $(x, y) \in \mathcal{Q} \times \mathbb{R}^m$, if we update s by $s = P_{\mathcal{Q}}(c - A^T y - x)$, then (x, y, s) is a solution of (5) as long as $Ax = b$ and $A^T y + s = c$. Hence, our main work is how to find $(x^{k+1}, y^{k+1}, s^{k+1})$ for the current (x^k, y^k, s^k), which can be done by the following non-interior point algorithm.

Algorithm 3.1 (A non-interior point algorithm for SOCPs).

Step 0. Choose $(x^0, y^0) \in \mathbb{R}^n \times \mathbb{R}^m$, $\gamma \in (0, 2)$, $\varepsilon > 0$. Set $k := 0$.

Step 1. Set $x^k := P_{\mathcal{Q}}(x^k)$, $s^k := P_{\mathcal{Q}}(c - A^T y^k - x^k)$.

Step 2. If $\|c - A^T y^k - s^k\|^2 + \|Ax^k - b\|^2 \leqslant \varepsilon$, then stop. Else, go to Step 3.

Step 3. Solve the following system of linear equations to obtain $\Delta w^k := (\Delta x^k, \Delta y^k) \in \mathbb{R}^n \times \mathbb{R}^m$:

$$\begin{pmatrix} I_n & -A^T \\ A & I_m \end{pmatrix} \begin{pmatrix} \Delta x^k \\ \Delta y^k \end{pmatrix} = -\gamma \begin{pmatrix} c - A^T y^k - s^k \\ Ax^k - b \end{pmatrix}. \tag{25}$$

Step 4. Set $x^{k+1} := x^k + \Delta x^k$, $y^{k+1} := y^k + \Delta y^k$ and $k := k + 1$. Go to Step 1.

Remark 3.1. The coefficient matrix

$$\begin{pmatrix} I_n & -A^T \\ A & I_m \end{pmatrix}$$

is invertible (in fact, it is positive definite by the Schur complement lemma [27]) for any $A \in \mathbb{R}^{m\times n}$, because the matrix AA^T is always positive semidefinite, and $I_m + AA^T$, i.e., the Schur complement of I_n, is always positive definite and hence invertible. Therefore, the linear system (25) has a unique solution. Taking into account Assumption 2.1, we conclude that Algorithm 3.1 is well defined.

R e m a r k 3.2. Straightforward evaluation yields

$$\begin{pmatrix} I_n & -A^T \\ A & I_m \end{pmatrix}^{-1} = \begin{pmatrix} (I_n + A^T A)^{-1} & A^T (I_m + AA^T)^{-1} \\ -A(I_n + A^T A)^{-1} & (I_m + AA^T)^{-1} \end{pmatrix},$$

from which we can see that, for large and sparse second-order cone programming problems, the main cost of solving (25) lies in inverting $I_n + A^T A$ and $I_m + AA^T$, which can be done efficiently via sparse Cholesky factorization.

4. Global convergence

In this section we discuss the global convergence property of Algorithm 3.1. To this end, we let Θ be the solution set of (5) and assume Θ is nonempty.

Lemma 4.1. *Let* $(x^k, y^k) \in \mathcal{Q} \times \mathbb{R}^m$, $s^k = P_{\mathcal{Q}}(c - A^T y^k - x^k)$. *Then for any* $w^* = (x^*, y^*, s^*) \in \Theta$ *we have*

$$(26)\qquad \begin{pmatrix} x^k - x^* \\ y^k - y^* \end{pmatrix}^T \begin{pmatrix} I_n & -A^T \\ A & I_m \end{pmatrix} \begin{pmatrix} c - A^T y^k - s^k \\ Ax^k - b \end{pmatrix} \geqslant \|Ax^k - b\|^2 + \|c - A^T y^k - s^k\|^2.$$

P r o o f. Since $w^* = (x^*, y^*, s^*) \in \Theta$, i.e., w^* satisfies (5), hence

$$x^* \succcurlyeq_{\mathcal{Q}} 0, \quad s^* \succcurlyeq_{\mathcal{Q}} 0, \quad x^* \circ s^* = 0$$

and

$$(27)\qquad \langle (x^*; y^*), (s^k - s^*; 0) \rangle = \langle x^*, s^k \rangle \geqslant 0.$$

On the other hand, choosing $v = c - A^T y^k - x^k$ and $\omega = s^*$ in (16), and taking into account $P_{\mathcal{Q}}(v) = s^k$, we get

$$\langle c - A^T y^k - x^k - s^k, s^k - s^* \rangle \geqslant 0.$$

Moreover, we have

$$(28)\qquad \langle (c - A^T y^k - x^k - s^k; y^k - Ax^k + b), (s^k - s^*; 0) \rangle \geqslant 0.$$

Adding (27) to (28) and letting $x^k - c + A^T y^k + s^k - x^* = \hat{x}$, $y^k - Ax^k + b - y^* = \hat{y}$ yields

$$(29)\qquad \langle (\hat{x}; \hat{y}), (s^* - s^k; 0) \rangle \geqslant 0.$$

Since $w^* \in \mathcal{Q}$, we can rewrite (29) as

$$\langle (c - A^T y^* - s^k; Ax^* - b), (\hat{x}; \hat{y}) \rangle \geqslant 0.$$

Rearranging the above inequality, we have

$$\begin{pmatrix} (x^k - x^*) - (c - A^T y^k - s^k) \\ (y^k - y^*) - (Ax^k - b) \end{pmatrix}^T \begin{pmatrix} A^T(y^k - y^*) + (c - A^T y^k - s^k) \\ A(x^* - x^k) + (Ax^k - b) \end{pmatrix} \geqslant 0.$$

Thus

$$\begin{aligned} &\begin{pmatrix} (x^k - x^*) - (c - A^T y^k - s^k) \\ (y^k - y^*) - (Ax^k - b) \end{pmatrix}^T \begin{pmatrix} \mathbf{0}_n & A^T \\ -A & \mathbf{0}_m \end{pmatrix} \begin{pmatrix} x^k - x^* \\ y^k - y^* \end{pmatrix} \\ &\quad + \begin{pmatrix} (x^k - x^*) - (c - A^T y^k - s^k) \\ (y^k - y^*) - (Ax^k - b) \end{pmatrix}^T \begin{pmatrix} c - A^T y^k - s^k \\ Ax^k - b \end{pmatrix} \geqslant 0. \end{aligned} \tag{30}$$

Furthermore, by taking into account

$$\begin{pmatrix} x^k - x^* \\ y^k - y^* \end{pmatrix}^T \begin{pmatrix} \mathbf{0}_n & A^T \\ -A & \mathbf{0}_m \end{pmatrix} \begin{pmatrix} x^k - x^* \\ y^k - y^* \end{pmatrix} = 0,$$

we have

$$\begin{aligned} &\begin{pmatrix} (x^k - x^*) - (c - A^T y^k - s^k) \\ (y^k - y^*) - (Ax^k - b) \end{pmatrix}^T \begin{pmatrix} \mathbf{0}_n & A^T \\ -A & \mathbf{0}_m \end{pmatrix} \begin{pmatrix} x^k - x^* \\ y^k - y^* \end{pmatrix} \\ &= \begin{pmatrix} c - A^T y^k - s^k \\ Ax^k - b \end{pmatrix}^T \begin{pmatrix} \mathbf{0}_n & -A^T \\ A & \mathbf{0}_m \end{pmatrix} \begin{pmatrix} x^k - x^* \\ y^k - y^* \end{pmatrix} \\ &= \begin{pmatrix} x^k - x^* \\ y^k - y^* \end{pmatrix}^T \begin{pmatrix} I_n & -A^T \\ A & I_m \end{pmatrix} \begin{pmatrix} c - A^T y^k - s^k \\ Ax^k - b \end{pmatrix} \\ &\quad - \begin{pmatrix} x^k - x^* \\ y^k - y^* \end{pmatrix}^T \begin{pmatrix} c - A^T y^k - s^k \\ Ax^k - b \end{pmatrix}, \end{aligned} \tag{31}$$

and

$$\begin{aligned} &\begin{pmatrix} (x^k - x^*) - (c - A^T y^k - s^k) \\ (y^k - y^*) - (Ax^k - b) \end{pmatrix}^T \begin{pmatrix} c - A^T y^k - s^k \\ Ax^k - b \end{pmatrix} \\ &= \begin{pmatrix} x^k - x^* \\ y^k - y^* \end{pmatrix}^T \begin{pmatrix} c - A^T y^k - s^k \\ Ax^k - b \end{pmatrix} \\ &\quad - (\|Ax^k - b\|^2 + \|c - A^T y^k - s^k\|^2). \end{aligned} \tag{32}$$

Substituting (31) and (32) into (30) and rearranging the inequality, we get the desired conclusion. □

The next result accounts for the termination rule used in Step 2. It can be easily obtained from Lemma 2.1.

Lemma 4.2. *Let* $w^k = (x^k, y^k, s^k)$ *be generated by Algorithm* 3.1. *Then we have*

$$\|\Phi(w^k)\|^2 \leqslant 2(\|Ax^k - b\|^2 + \|c - A^T y^k - s^k\|^2). \tag{33}$$

Lemma 4.3. *Let* w^k *be generated by Algorithm* 3.1. *Then for any* $w^* \in \Theta$ *we have*

$$\begin{aligned} &\left\| \begin{pmatrix} I_n & -A^T \\ A & I_m \end{pmatrix} \begin{pmatrix} x^{k+1} - x^* \\ y^{k+1} - y^* \end{pmatrix} \right\|^2 \\ &\quad \leqslant \left\| \begin{pmatrix} I_n & -A^T \\ A & I_m \end{pmatrix} \begin{pmatrix} x^k - x^* \\ y^k - y^* \end{pmatrix} \right\|^2 \\ &\qquad - \gamma(2-\gamma)(\|Ax^k - b\|^2 + \|c - A^T y^k - s^k\|^2), \end{aligned} \tag{34}$$

where $\gamma \in (0, 2)$.

P r o o f. From Step 3 we obtain

$$\begin{pmatrix} I_n & -A^T \\ A & I_m \end{pmatrix} \begin{pmatrix} x^{k+1} \\ y^{k+1} \end{pmatrix} = \begin{pmatrix} I_n & -A^T \\ A & I_m \end{pmatrix} \begin{pmatrix} x^k \\ y^k \end{pmatrix} - \gamma \begin{pmatrix} c - A^T y^k - s^k \\ Ax^k - b \end{pmatrix}. \tag{35}$$

Adding the following item

$$\begin{pmatrix} I_n & -A^T \\ A & I_m \end{pmatrix} \begin{pmatrix} -x^* \\ -y^* \end{pmatrix}$$

to both sides of (35) yields

$$\begin{pmatrix} I_n & -A^T \\ A & I_m \end{pmatrix} \begin{pmatrix} x^{k+1} - x^* \\ y^{k+1} - y^* \end{pmatrix} = \begin{pmatrix} I_n & -A^T \\ A & I_m \end{pmatrix} \begin{pmatrix} x^k - x^* \\ y^k - y^* \end{pmatrix} - \gamma \begin{pmatrix} c - A^T y^k - s^k \\ Ax^k - b \end{pmatrix}.$$

Therefore, by Lemma 4.1 we obtain

$$\begin{aligned} &\left\| \begin{pmatrix} I_n & -A^T \\ A & I_m \end{pmatrix} \begin{pmatrix} x^{k+1} - x^* \\ y^{k+1} - y^* \end{pmatrix} \right\|^2 \\ &\quad = \left\| \begin{pmatrix} I_n & -A^T \\ A & I_m \end{pmatrix} \begin{pmatrix} x^k - x^* \\ y^k - y^* \end{pmatrix} - \gamma \begin{pmatrix} c - A^T y^k - s^k \\ Ax^k - b \end{pmatrix} \right\|^2 \\ &\quad = \left\| \begin{pmatrix} I_n & -A^T \\ A & I_m \end{pmatrix} \begin{pmatrix} x^k - x^* \\ y^k - y^* \end{pmatrix} \right\|^2 + \gamma^2 \left\| \begin{pmatrix} c - A^T y^k - s^k \\ Ax^k - b \end{pmatrix} \right\|^2 \\ &\qquad - 2\gamma \begin{pmatrix} x^k - x^* \\ y^k - y^* \end{pmatrix} \begin{pmatrix} I_n & A^T \\ -A & I_m \end{pmatrix} \begin{pmatrix} c - A^T y^k - s^k \\ Ax^k - b \end{pmatrix} \\ &\quad \leqslant \left\| \begin{pmatrix} I_n & -A^T \\ A & I_m \end{pmatrix} \begin{pmatrix} x^k - x^* \\ y^k - y^* \end{pmatrix} \right\|^2 \\ &\qquad - \gamma(2-\gamma)(\|Ax^k - b\|^2 + \|c - A^T y^k - s^k\|^2), \end{aligned}$$

which completes the proof. □

From Lemma 4.2 and 4.3 we can easily obtain the following result.

Lemma 4.4. *Let $\{w^k\}$ be the sequence generated by Algorithm* 3.1. *Then we have*

$$\lim_{k\to+\infty}[\|Ax^k-b\|^2+\|c-A^Ty^k-s^k\|^2]=0,$$

and

$$\lim_{k\to+\infty}\Phi(w^k)=0.$$

We are now in the position to give the global convergence result for Algorithm 3.1.

Theorem 4.1. *Suppose that $\{w^k\}$ is any sequence generated by Algorithm* 3.1. *Then the following results hold.*

(a) *If Assumption* 2.1 *holds, $\{w^k\}$ is bounded and hence, it has at least one accumulation point $w^*=(x^*,y^*,s^*)$ with $\Phi(w^*)=0$ and $x^*,s^*\in\mathcal{Q}$.*

(b) *Every accumulation point of the sequence $\{w^k\}$ is a solution of the optimality conditions* (P) *and* (D).

Proof. (a) From Assumption 2.1 we know that (5) have solutions. Suppose that $\hat w=(\hat x,\hat y,\hat s)$ is a solution of (5). Since

$$\begin{aligned}\left\|\begin{pmatrix}I_n & -A^T\\ A & I_m\end{pmatrix}\begin{pmatrix}x^k-\hat x\\ y^k-\hat y\end{pmatrix}\right\|^2 &= \left\|\begin{pmatrix}x^k-\hat x\\ y^k-\hat y\end{pmatrix}+\begin{pmatrix}\mathbf{0}_n & -A^T\\ A & \mathbf{0}_m\end{pmatrix}\begin{pmatrix}x^k-\hat x\\ y^k-\hat y\end{pmatrix}\right\|^2\\ &= \left\|\begin{pmatrix}x^k-\hat x\\ y^k-\hat y\end{pmatrix}\right\|^2+\left\|\begin{pmatrix}\mathbf{0}_n & -A^T\\ A & \mathbf{0}_m\end{pmatrix}\begin{pmatrix}x^k-\hat x\\ y^k-\hat y\end{pmatrix}\right\|^2,\end{aligned}$$

from Lemma 4.3 we have

$$\begin{aligned}\left\|\begin{pmatrix}x^k-\hat x\\ y^k-\hat y\end{pmatrix}\right\|^2 &= \left\|\begin{pmatrix}I_n & -A^T\\ A & I_m\end{pmatrix}\begin{pmatrix}x^k-\hat x\\ y^k-\hat y\end{pmatrix}\right\|^2\\ &\quad-\left\|\begin{pmatrix}\mathbf{0}_n & -A^T\\ A & \mathbf{0}_m\end{pmatrix}\begin{pmatrix}x^k-\hat x\\ y^k-\hat y\end{pmatrix}\right\|^2\\ &\leqslant \left\|\begin{pmatrix}I_n & -A^T\\ A & I_m\end{pmatrix}\begin{pmatrix}x^k-\hat x\\ y^k-\hat y\end{pmatrix}\right\|^2\\ &\leqslant \left\|\begin{pmatrix}I_n & -A^T\\ A & I_m\end{pmatrix}\begin{pmatrix}x^0-\hat x\\ y^0-\hat y\end{pmatrix}\right\|^2,\end{aligned}\tag{36}$$

which means that the sequence $\{(x^k,y^k)\}$ is bounded. On the other hand, by the definition of s^k and continuity of the projection operator we know that $\{s^k\}$ is bounded. Hence, the sequence $\{(x^k,y^k,s^k)\}$ has at least one accumulation point.

(b) Let (x^*,y^*,s^*) be any accumulation of $\{(x^k,y^k,s^k)\}$ and without loss of generality, let us assume that the subsequence $\{(x^{k_i},y^{k_i},s^{k_i})\}$ converges to (x^*,y^*,s^*).

According to Lemmas 4.2 and 4.4 we have

$$\lim_{i\to+\infty} \Phi(x^{k_i}, y^{k_i}, s^{k_i}) = \Phi(x^*, y^*, s^*) = 0.$$

Therefore, (x^*, y^*, s^*) is a solution of (5). Due to the equivalence of (P), (D), and (5), we know that (x^*, y^*, s^*) is also a solution of (P) and (D). □

5. Preliminary numerical results

Now, we deal with numerical tests using Algorithm 3.1. All experiments were performed on a personal computer (IBM R40e) with 512 MB memory and Intel(R) Pentium(R) 4 CPU 2.00 GHz. The operating system was Windows XP (SP2) and the implementations were done in MATLAB 7.0.1 with double precision. We use $\|c - A^T y^k - s^k\|^2 + \|Ax^k - b\|^2 \leqslant 10^{-6}$ as the stopping rule. The numerical results are summarized in Tab. 1 for different tested problems. In Tab. 1, **Iter** denotes the number of iterations. CPU time denotes the time needed for obtaining the optimal solution satisfying the stopping rule, **FV** denotes the value of $\|c - A^T y^k - s^k\|^2 + \|Ax^k - b\|^2$ at the final iterate. In the sequel, we give a brief description of the tested problems.

We consider the SOCP problem $\min\{\langle c, x\rangle\colon\ Ax = b,\ x \in \mathcal{Q}\}$, whose data are given as follows:

Problem 1.

$$A = \begin{pmatrix} 2 & 1 \\ 1 & -1 \end{pmatrix}, \quad c = (2; 1), \quad b = (2; 1).$$

The optimal solution is $x^* = (1; 0)$, $y^* = (1; 0)$, and the optimal value of the objective function is 2.

Problem 2.

$$A = \begin{pmatrix} 2 & 1 \\ 1 & -1 \\ 4 & 2 \end{pmatrix}, \quad c = (2; 1), \quad b = (2; 1; 4).$$

The optimal solution is $x^* = (1; 0)$, $y^* = (0.2; 0; 0.4)$, and the optimal value of the objective function is again 2.

Problem 3.

$$A = \begin{pmatrix} 10 & 2 & & & \\ -2 & 10 & 2 & & \\ & \ddots & \ddots & \ddots & \\ & & -2 & 10 & 2 \\ & & & -2 & 10 \end{pmatrix} \in \mathbb{R}^{m\times n},$$

$$c = 100e_n + 4\,\mathbf{rand}(n,1) - 2\,\mathbf{ones}(n,1),$$

$$b = 100e_m + 4\,\mathbf{rand}(m,1) - 2\,\mathbf{ones}(m,1),$$

where $\mathbf{ones}(k,1)$ denotes the vector with dimension k whose all elements are ones, and $\mathbf{rand}(n,1)$ is an n-dimensional real vector with random entries, chosen from a uniform distribution on the interval $(0,1)$.

Problem 4.

$$B = \begin{pmatrix} 10 & 2 & & & \\ -2 & 10 & 2 & & \\ & \ddots & \ddots & \ddots & \\ & & -2 & 10 & 2 \\ & & & -2 & 10 \end{pmatrix} \in \mathbb{R}^{m\times m}, \quad A = [B, \mathbf{randn}(m, n-m)],$$

$$c = 100e_n + 4\,\mathbf{rand}(n,1) - 2\,\mathbf{ones}(n,1),$$

$$b = 100e_m + 4\,\mathbf{rand}(m,1) - 2\,\mathbf{ones}(m,1),$$

where $\mathbf{randn}(m, n-m)$ is an m-by-$(n-m)$ real matrix with random entries, chosen from the standard normal distribution on the interval $(0,1)$, $[B, \mathbf{randn}(m, n-m)]$ is the block matrix obtained by adjoining the matrices B and $\mathbf{randn}(m, n-m)$ in a row.

Note that in Problem 1, the row vectors of A are linearly independent, while in the case of Problem 2, A is not a full row rank matrix.

Displayed in Tab. 1 are numerical results for Algorithm 3.1 for the above four problems. The computational results show that the present method is efficient as far as the numerical results are considered. Moreover, it can also deal with the case that A has not the full row rank. Furthermore, it can deal with large-scale and sparse second-order cone programming. Therefore, the new method may be of practical interest.

Problem	m	n	γ	x_0	y_0	**Iter**	CPU time (s)	**FV**
Problem 1	2	2	0.9	e_2	$-e_2$	11	0.09	2.93×10^{-7}
	2	2	0.9	$0.5e_2$	0	10	0.06	6.65×10^{-7}
	2	2	1	0	0	9	0.10	2.20×10^{-7}
	2	2	1.5	$-e_2$	$0.5e_2$	15	0.02	8.66×10^{-7}
	2	2	0.9	$-0.5e_2$	0	10	0.08	7.02×10^{-7}
	2	2	1.5	$-0.5e_2$	$-e_2$	17	0.08	4.13×10^{-7}
Problem 2	3	2	0.8	e_2	0	10	0.08	6.85×10^{-7}
	3	2	1	$0.5e_2$	$-e_3$	9	0.03	3.60×10^{-7}
	3	2	0.9	0	0	9	0.08	9.93×10^{-7}
	3	2	0.9	$-0.5e_2$	$0.5e_3$	10	0.01	4.27×10^{-7}
	3	2	1.6	$-0.5e_2$	0	14	0.09	1.59×10^{-7}
	3	2	1.2	$-e_2$	$-e_3$	8	0.05	5.39×10^{-7}
Problem 3	150	150	0.9	0	0	38	4.907	6.6852×10^{-8}
	200	200	1.0	$\mathbf{ones}(200, 1)$	0	18	3.845	4.3319×10^{-7}
	200	200	1.5	$\mathbf{ones}(200, 1)$	$\mathbf{ones}(200, 1)$	33	6.790	8.6068×10^{-7}
Problem 4	150	200	1.6	0	0	32	4.537	2.2935×10^{-7}
	150	200	1.4	0	$\mathbf{ones}(150, 1)$	35	4.717	9.3174×10^{-7}
	150	200	1.8	$\mathbf{ones}(200, 1)$	$\mathbf{ones}(150, 1)$	49	6.530	9.2915×10^{-7}

Table 1. Numerical results for Algorithm 3.1.

6. Final remarks

In the paper a non-interior point algorithm for SOCP problems based on projection is proposed and analyzed. The method provides a stronger convergence result than that for IPMs and the smoothing methods. It is versatile and easy to implement. Preliminary numerical results demonstrate that the algorithm given in this paper is effective and has good numerical performance for second-order cone programming problems, especially, in the following cases: the row vectors of A are not linearly independent, there is no strict complementarity and the SOCPs concerned have large-scale sparse structure. How to improve the method to obtain local convergence and further numerical tests comparing our algorithm with existing methods will be the topic of future research.

Acknowledgements. The authors would like to thank the anonymous referees for their valuable comments and suggestions.

References

[1] *F. Alizadeh, D. Goldfarb*: Second-order cone programming. Math. Program. *95* (2003), 3–51.

[2] *H. Y. Benson, R. J. Vanderbei*: Solving problems with semidefinite and related constraints using interior-point methods for nonlinear programming. Math. Program. Ser. B *95* (2003), 279–302.

[3] *A. Ben-Tal, A. Nemirovski*: On polyhedral approximations of the second order cone. Math. Oper. Res. *26* (2001), 193–205.

[4] *X. D. Chen, D. Sun, J. Sun*: Complementarity functions and numerical experiments on some smoothing Newton methods for second-order-cone complementarity problems. Comput. Optim. Appl. *25* (2003), 39–56.

[5] *I. C. Choi, C. L. Monma, D. Shanno*: Further developments of primal-dual interior point methods. ORSA J. Comput. *2* (1990), 304–311.

[6] *J. Faraut, A. Korányi*: Analysis on Symmetric Cones. Oxford University Press, Oxford, 1994.

[7] *L. Faybusovich*: Euclidean Jordan algebras and interior-point algorithms. Positivity *1* (1997), 331–357.

[8] *L. Faybusovich*: Linear systems in Jordan algebras and primal-dual interior point algorithms. J. Comput. Appl. Math. *86* (1997), 149–175.

[9] *M. Fukushima, Z.-Q. Luo, P. Tseng*: Smoothing functions for second-order cone complementarity problems. SIAM J. Optim. *12* (2002), 436–460.

[10] *Q. M. Han*: Projection and contraction methods for semidefinite programming. Appl. Math. Comput. *95* (1998), 275–289.

[11] *S. Hayashi, N. Yamashita, M. Fukushima*: A combined smoothing and regularization method for monotone second-order cone complementarity problems. SIAM J. Optim. *15* (2005), 593–615.

[12] *B. S. He, L. Z. Liao, D. R. Han, H. Yang*: A new inexact alternating directions method for monotone variational inequality. Math. Program. *92* (2002), 103–118.

[13] *Z. Huang, W. Gu*: A smoothing-type algorithm for solving linear complementarity problems with strong convergence properties. Appl. Math. Optim. *57* (2008), 17–29.

[14] *S. Kim, M. Kojima*: Second order cone programming relaxations of nonconvex quadratic optimization problems. Optim. Methods Softw. *15* (2001), 201–224.

[15] *Y. J. Kuo, H. D. Mittelmann*: Interior point methods for second-order cone programming and OR applications. Comput. Optim. Appl. *28* (2004), 255–285.

[16] *Y. Liu, L. Zhang, Y. Wang*: Analysis of a smoothing method for symmetric conic linear programming. J. Appl. Math. Comput. *22* (2006), 133–148.

[17] *M. S. Lobo, L. Vandenberghe, S. Boyd, H. Lebret*: Applications of second order cone programming. Linear Algebra Appl. *284* (1998), 193–228.

[18] *R. D. C. Monteiro, T. Tsuchiya*: Polynomial convergence of primal-dual algorithms for the second-order cone program based on the MZ-family of directions. Math. Program. *88* (2000), 61–83.

[19] *R. D. C. Monteiro, Y. Zhang*: A unified analysis for a class of path-following primal-dual interior-point algorithms for semidefinite programming. Math. Program. *81* (1998), 281–299.

[20] *Yu. E. Nesterov, A. S. Nemirovskii*: Interior-Point Polynomial Algorithms in Convex Programming. SIAM, Philadelphia, 1994.

[21] *B. K. Rangarajan*: Polynomial convergence of infeasible-interior-point methods over symmetric cones. SIAM J. Optim. *16* (2006), 1211–1229.

[22] *B. Rangarajan, M. J. Todd*: Convergence of infeasible-interior-point methods for self-scaled conic programming. Technical Report 1388. School of OR & IE, Cornell University.
[23] *S. H. Schmieta, F. Alizadeh*: Extension of primal-dual interior-point algorithm to symmetric cones. Math. Program. *96* (2003), 409–438.
[24] *S. H. Schmieta, F. Alizadeh*: Associative and Jordan algebras, and polynomial time interior-point algorithms for symmetric cones. Math. Oper. Res. *26* (2001), 543–564.
[25] *S. H. Schmieta, F. Alizadeh*: Extension of commutative class of primal-dual interior point algorithms to symmetric cones. Technical Report RRR 13-99, RUTCOR. Rutgers University, August 1999.
[26] *D. Sun, J. Sun*: Strong semismoothness of the Fischer-Burmeister SDC and SOC complementarity functions. Math. Program., Ser. A *103* (2005), 575–581.
[27] *M. Todd*: Semidefinite optimization. Acta Numerica *10* (2001), 515–560.
[28] *K. C. Toh, M. J. Todd, R. H. Tütüncü*: SDPT3-A Matlab software package for semidefinite programming. Version 2.1. Optim. Methods Softw. *11* (1999), 545–581.
[29] *T. Tsuchiya*: A convergence analysis of the scaling-invariant primal-dual path-following algorithms for second-order cone programming. Optim. Methods Softw. *11* (1999), 141–182.
[30] *T. Tsuchiya*: A polynomial primal-dual path-following algorithm for second-order cone programming. Technical Report No. 649. The Institute of Statistical Mathematics, Tokyo, 1997.
[31] *S. L. Wang, H. Yang, B. S. He*: Solving a class of asymmetric variational inequalities by a new alternating direction method. Comput. Math. Appl. *40* (2000), 927–937.
[32] *S. J. Wright*: Primal-Dual Interior Point Methods. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, 1997.
[33] *G. L. Xue, Y. Y. Ye*: An efficient algorithm for minimizing a sum of Euclidean norms with applications. SIAM J. Optim. *7* (1997), 1017–1036.
[34] *Y. Zhang*: On extending some primal-dual interior-point algorithms from linear programming to semidefinite programming. SIAM J. Optim. *8* (1998), 365–386.

Authors' addresses: *L. Fang* (corresponding author), College of Mathematics and System Science, Taishan University, 271021 Tai'an, P. R. China, e-mail: `fangliang3@163.com`, and Department of Mathematics, Shanghai Jiao Tong University, 200240 Shanghai, P. R. China, e-mail: `fangliang3@hotmail.com`; *Guoping He*, College of Information Science and Engineering, Shandong University of Science and Technology, 266510 Qingdao, P. R. China, and Department of Mathematics, Shaghai Jiao Tong University, 200240 Shanghai, P. R. China, e-mail: `hegp@263.net`; *Li Sun*, College of Information Science and Engineering, Shandong Agricultural University, 271018 Tai'an, P. R. China, e-mail: `sunlishi @hotmail.com`.