Jingyong Tang; Guoping He; Li Dong; Liang Fang; Jinchuan Zhou

A smoothing Newton method for the second-order cone complementarity problem

Applications of Mathematics, Vol. 58 (2013), No. 2, 223–247

Persistent URL: `http://dml.cz/dmlcz/143164`

A SMOOTHING NEWTON METHOD FOR THE SECOND-ORDER CONE COMPLEMENTARITY PROBLEM

Jingyong Tang, Xinyang and Shanghai, Guoping He, Qingdao,
Li Dong, Xinyang, Liang Fang, Tai'an, Jinchuan Zhou, Zibo

(Received September 12, 2011)

Abstract. In this paper we introduce a new smoothing function and show that it is coercive under suitable assumptions. Based on this new function, we propose a smoothing Newton method for solving the second-order cone complementarity problem (SOCCP). The proposed algorithm solves only one linear system of equations and performs only one line search at each iteration. It is shown that any accumulation point of the iteration sequence generated by the proposed algorithm is a solution to the SOCCP. Furthermore, we prove that the generated sequence is bounded if the solution set of the SOCCP is nonempty and bounded. Under the assumption of nonsingularity, we establish the local quadratic convergence of the algorithm without the strict complementarity condition. Numerical results indicate that the proposed algorithm is promising.

Keywords: second-order cone complementarity problem, smoothing function, smoothing Newton method, global convergence, quadratic convergence

MSC 2010: 90C25, 90C30, 90C33

1. Introduction

The second-order cone (SOC) in $\mathbb{R}^n$ ($n \geqslant 1$), also called the Lorentz cone, is defined as

$$\mathcal{K}^n := \{(x_1, \bar{x}^{\mathrm{T}})^{\mathrm{T}} \in \mathbb{R} \times \mathbb{R}^{n-1} \colon x_1 \geqslant \|\bar{x}\|\}.$$

Here and below, $\|\cdot\|$ denotes the 2-norm defined by $\|x\| = \sqrt{x^{\mathrm{T}}x}$ for a vector x. For convenience, we write $(u^{\mathrm{T}}, v^{\mathrm{T}})^{\mathrm{T}}$ as (u, v) for any vectors $u, v \in \mathbb{R}^n$ throughout the paper.

This paper was partly supported by the Project of Shandong Province Higher Educational Science and Technology Program (J10LA51), National Natural Science Foundation of China (10971122, 11101248), and Excellent Young Scientist Foundation of Shandong Province (BS2011SF024).

We consider the second-order cone complementarity problem (SOCCP) as follows:

$$\text{Find } (x,y)\in\mathbb{R}^n\times\mathbb{R}^n, \text{ such that } x\in\mathcal{K},\ y\in\mathcal{K},\ x^{\mathrm{T}}y=0,\ y=F(x), \tag{1.1}$$

where $F\colon \mathbb{R}^n\to\mathbb{R}^n$ is a continuously differentiable function, and $\mathcal{K}\subset\mathbb{R}^n$ is the Cartesian product of second-order cones, that is,

$$\mathcal{K}=\mathcal{K}^{n_1}\times\ldots\times\mathcal{K}^{n_r},$$

with $r,n_1,\ldots,n_r\geqslant 1$ and $n=\sum\limits_{i=1}^{r}n_i$, and $x=(x^1,\ldots,x^r)$, $y=(y^1,\ldots,y^r)$ with $x^i,y^i\in\mathcal{K}^{n_i}$, $i=1,\ldots,r$.

The following proposition shows that the complementarity condition on $\mathcal{K}=\mathcal{K}^{n_1}\times\ldots\times\mathcal{K}^{n_r}$ can be decomposed into complementarity conditions on each $\mathcal{K}^{n_i}$.

Proposition 1.1 ([17]). *Let $\mathcal{K}=\mathcal{K}^{n_1}\times\ldots\times\mathcal{K}^{n_r}$, $x=(x^1,\ldots,x^r)\in\mathbb{R}^{n_1}\times\ldots\times\mathbb{R}^{n_r}$ and $y=(y^1,\ldots,y^r)\in\mathbb{R}^{n_1}\times\ldots\times\mathbb{R}^{n_r}$. Then the following relation holds:*

$$x\in\mathcal{K},\ \ y\in\mathcal{K},\ \ x^{\mathrm{T}}y=0\Leftrightarrow x^i\in\mathcal{K}^{n_i},\ \ y^i\in\mathcal{K}^{n_i},\ \ (x^i)^{\mathrm{T}}y^i=0,\ \ i=1,\ldots,r.$$

In the following analysis, we assume that $\mathcal{K}=\mathcal{K}^n$. We do not lose any generality, because in view of Proposition 1.1, our analysis can be extended to the general case in a straightforward manner.

The SOCCP contains a wide class of problems such as the nonlinear complementarity problem (NCP) and the second-order cone programming (SOCP) [17]. A number of methods for solving the SOCCP have been proposed. They include the smoothing-regularization method [17], the derivative-free descent method [23], the merit function method (e.g., [6], [8]), the damped Gauss-Newton method [22], the nonsmooth method [7] and so on. Most of the methods are proposed for the monotone SOCCP.

Definition 1.2 ([17]). The function $F\colon \mathbb{R}^n\to\mathbb{R}^n$ is said to be monotone if for any $(x,y)\in\mathbb{R}^n\times\mathbb{R}^n$,

$$(x-y)^{\mathrm{T}}(F(x)-F(y))\geqslant 0.$$

Recently, smoothing Newton methods have attracted a lot of attention partially due to their encouraging convergent properties and numerical results. The main idea of this class of methods is to reformulate the problem concerned as a family of parameterized smooth equations and then to solve the smooth equations approximately by using Newton's method at each iteration. By driving the parameter to converge to zero, one can expect to find a solution to the original problem. However, in order to obtain the local superlinear (quadratic) convergence, some algorithms (e.g., [1], [2], [5], [26]) depend on the assumptions of uniform nonsingularity and strict complementarity. Recently, Qi, Sun and Zhou [27] proposed a class of smoothing Newton methods for the NCP and box constrained variational inequalities. The Qi-Sun-Zhou method [27] was shown to be locally superlinearly/quadratically convergent without strict complementarity. Due to its simplicity and weaker assumptions imposed on smoothing functions, the Qi-Sun-Zhou method [27] has been further studied for the NCP (e.g., [14], [20], [31], [32]) and the SOCP (e.g., [10]–[12], [15], [28]).

In this paper, we present a new smoothing function which is coercive under suitable assumptions. Based on this new function, we propose a smoothing Newton algorithm for the SOCCP by modifying and extending the Qi-Sun-Zhou method [27]. It is proved that the proposed algorithm has the following nice properties:

(a) It is well-defined and a solution to the SOCCP can be obtained from any accumulation point of the iteration sequence generated by this method.

(b) If the solution set of (1.1) is nonempty and bounded, then the iteration sequence is bounded and hence it has at least one accumulation point.

(c) The algorithm solves only one system of linear equations and performs only one line search per iteration.

(d) The whole iteration sequence converges to the accumulation point globally. Furthermore, if the Jacobian of F is Lipschitz continuous on $\mathbb{R}^n$, then the iteration sequence converges locally quadratically without strict complementarity.

The paper is organized as follows. In the next section, we introduce some preliminaries to be used in the subsequent sections. Moreover, we present the new smoothing function and give its properties. In Section 3, we propose a smoothing Newton method for solving the SOCCP and show the well-definedness of the algorithm. The global convergence and local quadratic convergence of the algorithm are investigated in Section 4. Preliminary numerical results are reported in Section 5. Some conclusions are made in Section 6.

In our notation, $\mathbb{R}^n$ denotes the space of n-dimensional real column vectors, and $\mathbb{R}^n_+$ (or $\mathbb{R}^n_{++}$) denotes the non-negative (or positive) orthant in $\mathbb{R}^n$. $\mathbb{R}^{n_1} \times \ldots \times \mathbb{R}^{n_r}$ is identified with $\mathbb{R}^{n_1+\cdots+n_r}$. Thus, $(x_1, \ldots, x_r) \in \mathbb{R}^{n_1} \times \ldots \times \mathbb{R}^{n_r}$ is viewed as a column vector in $\mathbb{R}^{n_1+\cdots+n_r}$. I denotes the identity matrix with suitable dimension. $\langle \cdot, \cdot \rangle$ denotes the Euclidean inner product. For any $x, y \in \mathbb{R}^n$, we write $x \succeq_{\mathcal{K}} y$ (or $x \succ_{\mathcal{K}} y$) if $x - y \in \mathcal{K}$ (or $x - y \in \operatorname{int} \mathcal{K}$, where $\operatorname{int} \mathcal{K}$ denotes the interior of $\mathcal{K}$). For any $\alpha, \beta > 0$, $\alpha = O(\beta)$ (or $\alpha = o(\beta)$) means that α/β is uniformly bounded (or tends to zero) as $\beta \to 0$.

2. Preliminaries and a new smoothing function

In this section, we first give a brief description of the Euclidean Jordan algebra associated with the SOC $\mathcal{K}$, which is a basic tool used in this paper. Then, we introduce a new smoothing function and give its properties.

For any vectors $x = (x_1, \bar{x}), s = (s_1, \bar{s}) \in \mathbb{R}\times\mathbb{R}^{n-1}$, their Jordan product associated with the SOC $\mathcal{K}$ is defined by

$$x \circ s := (x^{\mathrm{T}} s,\ x_1\bar{s} + s_1\bar{x}).$$

The identity element under this product is $\mathbf{e} := (1, 0, \ldots, 0)^{\mathrm{T}} \in \mathbb{R}^n$. Given an element $x = (x_1, \bar{x}) \in \mathbb{R} \times \mathbb{R}^{n-1}$, we define the symmetric matrix

$$L_x := \begin{pmatrix} x_1 & \bar{x}^{\mathrm{T}} \\ \bar{x} & x_1 I \end{pmatrix},$$

where I represents the $(n-1) \times (n-1)$ identity matrix. It is easy to verify that $x \circ s = L_x s = L_s x$ for any $x, s \in \mathbb{R}^n$. Moreover, L_x is positive semidefinite (or positive definite and hence nonsingular) if and only if $x \in \mathcal{K}$ (or $x \in \operatorname{int}\mathcal{K}$).

Spectral decomposition is one of the basic concepts in Euclidean Jordan algebras. For any $x = (x_1, \bar{x}) \in \mathbb{R} \times \mathbb{R}^{n-1}$, its spectral decomposition with respect to the SOC $\mathcal{K}$ is defined as

$$x = \lambda_1 u_1 + \lambda_2 u_2,$$

in which

$$\lambda_i = x_1 + (-1)^i \|\bar{x}\|$$

and

$$u_i = \begin{cases} \dfrac{1}{2}\Big(1, (-1)^i \dfrac{\bar{x}}{\|\bar{x}\|}\Big) & \text{if } \bar{x} \neq 0, \\ \dfrac{1}{2}\Big(1, (-1)^i \dfrac{\kappa}{\|\kappa\|}\Big) & \text{otherwise, for any } \kappa \neq 0, \end{cases} \qquad i = 1, 2.$$

Since $x \in \mathcal{K}$ (or $x \in \operatorname{int}\mathcal{K}$) if and only if both λ_1 and λ_2 are nonnegative (or positive), one can define

$$\begin{aligned} \sqrt{x} &= \sqrt{\lambda_1}u_1 + \sqrt{\lambda_2}u_2 && \text{for any } x \in \mathcal{K}, \\ x^{-1} &= \lambda_1^{-1}u_1 + \lambda_2^{-1}u_2 && \text{for any } x \in \operatorname{int}\mathcal{K}, \\ x^2 &= \lambda_1^2 u_1 + \lambda_2^2 u_2 && \text{for any } x \in \mathbb{R}^n. \end{aligned}$$

Note that $x \circ x^{-1} = \mathbf{e}$. Moreover, for any $x \in \mathbb{R}^n$, $x^2 = x \circ x$ and $x^2 \in \mathcal{K}$.

For any $x \in \mathbb{R}^n$, we define $[x]_+$ to be the nearest-point (in the Euclidean norm) projection of x onto $\mathcal{K}$. For any $\alpha \in \mathbb{R}$, let $[\alpha]_+ = \max\{0, \alpha\}$. Then from Proposition 3.3 in [16] we know that

$$|x| = \sqrt{x^2} = |\lambda_1| u_1 + |\lambda_2| u_2$$

and the projection of x onto $\mathcal{K}$ can be written as

$$[x]_+ = [\lambda_1]_+ u_1 + [\lambda_2]_+ u_2 = (x + |x|)/2.$$

Define the vector-valued function by

$$\varphi_0(x, y) := 2(x - [x - y]_+) = x + y - \sqrt{(x - y)^2}. \tag{2.1}$$

In [16], it has been shown that $\varphi_0(x, y)$ satisfies the following property:

$$\varphi_0(x, y) = 0 \Leftrightarrow x \in \mathcal{K}, \quad y \in \mathcal{K}, \ x^{\mathrm{T}} y = 0. \tag{2.2}$$

It is well known that φ_0 is typically nonsmooth, because it is not differentiable at $(0, 0) \in \mathbb{R}^n \times \mathbb{R}^n$ which limits its practical applications. In order to overcome this difficulty, we can use a smoothing function of φ_0.

Definition 2.1 ([17])**.** For a nondifferentiable function $g\colon \mathbb{R}^n \to \mathbb{R}^n$, a function $g_\mu\colon \mathbb{R}^n \to \mathbb{R}^n$ with a parameter $\mu > 0$ is called a smoothing function of g if

(i) g_μ is differentiable for any $\mu > 0$;

(ii) $\lim\limits_{\mu \downarrow 0} g_\mu(x) = g(x)$ for any $x \in \mathbb{R}^n$.

In this paper, by smoothing the symmetric perturbed function of φ_0, we obtain a new vector-valued function $\varphi(\mu, x, y)\colon \mathbb{R}_+ \times \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^n$ defined by

$$\varphi(\mu, x, y) = (\cos\mu + \sin\mu)(x + y) - \sqrt{(\cos\mu - \sin\mu)^2 (x - y)^2 + 4\mu^2 \mathbf{e}}, \tag{2.3}$$

where μ is a real parameter. As we will show, the function φ given in (2.3) is a smoothing function of φ_0 and it possesses some nice properties.

Theorem 2.2. *Let* $\varphi(\mu,x,y)\colon \mathbb{R}_+\times\mathbb{R}^n\times\mathbb{R}^n\to\mathbb{R}^n$ *be defined by* (2.3). *Denote*

$$\omega := \omega(\mu,x,y) = \sqrt{(\cos\mu-\sin\mu)^2(x-y)^2+4\mu^2\mathbf{e}}. \tag{2.4}$$

Then the following results hold:

(i) $\varphi(\mu,x,y)$ *is continuously differentiable at any* $(\mu,x,y)\in\mathbb{R}_{++}\times\mathbb{R}^n\times\mathbb{R}^n$ *with its Jacobian*

$$\varphi'(\mu,x,y) = \begin{pmatrix} (\cos\mu-\sin\mu)(x+y)-L_\omega^{-1}[-(1-2\sin^2\mu)(x-y)^2+4\mu\mathbf{e}] \\ (\cos\mu+\sin\mu)I-(\cos\mu-\sin\mu)^2L_\omega^{-1}L_{x-y} \\ (\cos\mu+\sin\mu)I+(\cos\mu-\sin\mu)^2L_\omega^{-1}L_{x-y} \end{pmatrix}. \tag{2.5}$$

(ii) $\varphi(\mu,x,y)$ *is a smoothing function of* $\varphi_0(x,y)$.

Proof. It is easy to show that $\varphi(\mu,x,y)$ is continuously differentiable at any $(\mu,x,y)\in\mathbb{R}_{++}\times\mathbb{R}^n\times\mathbb{R}^n$. For any $(\mu,x,y)\in\mathbb{R}_{++}\times\mathbb{R}^n\times\mathbb{R}^n$, it follows from (2.4) that $\omega\in\operatorname{int}\mathcal{K}$ and therefore L_ω is invertible. From the definition of ω, we get

$$\omega^2=(\cos\mu-\sin\mu)^2(x-y)^2+4\mu^2\mathbf{e}.$$

Then,

$$\begin{aligned}\omega'_\mu &= L_\omega^{-1}[-(1-2\sin^2\mu)(x-y)^2+4\mu\mathbf{e}],\\ \omega'_x &= (\cos\mu-\sin\mu)^2L_\omega^{-1}L_{x-y},\\ \omega'_y &= -(\cos\mu-\sin\mu)^2L_\omega^{-1}L_{x-y}.\end{aligned}$$

Therefore, due to the definition of φ, we have

$$\varphi'_\mu(\mu,x,y)=(\cos\mu-\sin\mu)(x+y)-L_\omega^{-1}[-(1-2\sin^2\mu)(x-y)^2+4\mu\mathbf{e}], \tag{2.6}$$

$$\varphi'_x(\mu,x,y)=(\cos\mu+\sin\mu)I-(\cos\mu-\sin\mu)^2L_\omega^{-1}L_{x-y}, \tag{2.7}$$

$$\varphi'_y(\mu,x,y)=(\cos\mu+\sin\mu)I+(\cos\mu-\sin\mu)^2L_\omega^{-1}L_{x-y}. \tag{2.8}$$

From (2.6)–(2.8), we have the desired Jacobian formula. Now we prove (ii). For any $x=(x_1,\bar{x})$ and $y=(y_1,\bar{y})\in\mathbb{R}\times\mathbb{R}^{n-1}$, by the spectral factorization of ω^2 we have

$$\varphi(\mu,x,y)=(\cos\mu+\sin\mu)(x+y)-\big(\sqrt{\lambda_1(\mu)}u_1(\mu)+\sqrt{\lambda_2(\mu)}u_2(\mu)\big)$$

where

$$\begin{aligned}\lambda_i(\mu) &= (\cos\mu-\sin\mu)^2\|x-y\|^2+4\mu^2+2(-1)^i\|v(\mu)\|, \quad i=1,2,\\ u_i(\mu) &= \begin{cases}\dfrac12\Big(1,(-1)^i\dfrac{v(\mu)}{\|v(\mu)\|}\Big) & \text{if } v(\mu)\neq0,\\ \dfrac12\Big(1,(-1)^i\dfrac{\kappa}{\|\kappa\|}\Big) & \text{otherwise, for any } \kappa\neq0,\end{cases} \quad i=1,2,\\ v(\mu) &= (\cos\mu-\sin\mu)^2(x_1-y_1)(\bar{x}-\bar{y}).\end{aligned}$$

In a similar way, we have

$$\varphi_0(x,y) = x + y - \big(\sqrt{\lambda_1}u_1 + \sqrt{\lambda_2}u_2\big),$$

where

$$\begin{aligned} \lambda_i &= \|x-y\|^2 + 2(-1)^i\|v\|, \quad i=1,2, \\ u_i &= \begin{cases} \dfrac{1}{2}\Big(1,(-1)^i\dfrac{v}{\|v\|}\Big) & \text{if } v \neq 0, \\ \dfrac{1}{2}\Big(1,(-1)^i\dfrac{\kappa}{\|\kappa\|}\Big) & \text{otherwise, for any } \kappa \neq 0, \end{cases} \qquad i=1,2, \\ v &= (x_1-y_1)(\bar{x}-\overline{y}). \end{aligned}$$

Without loss of generality, we choose the same $\kappa \in \mathbb{R}^{n-1}$ as in $u_i(\mu)$. It is obvious that $\lim\limits_{\mu\downarrow 0} v(\mu) = v$. Thus, we have

$$\lim_{\mu\downarrow 0}\lambda_i(\mu) = \lambda_i, \ \ \lim_{\mu\downarrow 0} u_i(\mu) = u_i, \quad i=1,2,$$

which implies that

$$\lim_{\mu\downarrow 0}\varphi(\mu,x,y) = \varphi_0(x,y).$$

Therefore, it follows from (i) and Definition 2.1 that $\varphi(\mu,x,y)$ is a smoothing function of $\varphi_0(x,y)$. The proof is completed. $\square$

At the end of this section, we discuss the coerciveness of the function φ defined by (2.3).

Theorem 2.3. *Let $\varphi(\mu,x,y)$ be defined by* (2.3), *and $\xi,\zeta \in \mathbb{R}_{++}$ with $\xi < \zeta$. Suppose that $\{(\mu_k,x^k,y^k)\} \subset \mathbb{R}_{++}\times\mathbb{R}^n\times\mathbb{R}^n$ is a sequence satisfying*

(a) *$\mu_k \in [\xi,\zeta]$, $\{(x^k,y^k)\}$ is unbounded; and*

(b) *there is a bounded sequence $\{(u^k,v^k)\}$ such that $\{\langle x^k-u^k, y^k-v^k\rangle\}$ is bounded below.*

Then $\{\varphi(\mu_k,x^k,y^k)\}$ is unbounded.

Proof. By using Lemma 4.2 in [19] and the fact that

$$\begin{aligned} \varphi &= (\cos\mu x + \sin\mu y) + (\sin\mu x + \cos\mu y) \\ &\quad - \sqrt{[(\cos\mu x + \sin\mu y) - (\sin\mu x + \cos\mu y)]^2 + 4\mu^2\mathbf{e}}, \end{aligned}$$

we can prove the theorem similarly as Theorem 4.1 in [19]. For brevity, we omit the details here. $\square$

3. Description of the algorithm

The aim of this section is to propose a smoothing Newton method for solving the SOCCP. Under suitable assumptions, we show the well-definedness of our algorithm.

Let $z := (\mu, x, y)$. We define the function $H(z)\colon \mathbb{R}_{++} \times \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}_{++} \times \mathbb{R}^n \times \mathbb{R}^n$ by

$$H(z) := \begin{pmatrix} \mu \\ F(x) - y \\ \varphi(\mu, x, y) \end{pmatrix}, \tag{3.1}$$

where φ is given by (2.3). Then, from Theorem 2.2 and (2.2), we know that (x, y) is the solution to the SOCCP if and only if $H(z) = 0$. Therefore, instead of solving the SOCCP, we may apply Newton's methods to solve the system of equations $H(z) = 0$.

Lemma 3.1 ([30]). *Let $a, b, u, v \in \mathbb{R}^n$ with $a \succ_{\mathcal{K}} 0$, $b \succ_{\mathcal{K}} 0$, $a \circ b \succ_{\mathcal{K}} 0$. If $\langle u, v\rangle \geqslant 0$ and $a \circ u + b \circ v = 0$, then $u = v = 0$.*

Theorem 3.2. *Let $z := (\mu, x, y)$ and $H(z)$ be defined by (3.1). Then the following results hold:*

(i) *$H(z)$ is continuously differentiable at any $z := (\mu, x, y) \in \mathbb{R}_{++} \times \mathbb{R}^n \times \mathbb{R}^n$ with its Jacobian*

$$H'(z) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & F'(x) & -I \\ \varphi'_\mu(z) & \varphi'_x(z) & \varphi'_y(z) \end{pmatrix}, \tag{3.2}$$

where

$$\varphi'_\mu(z) = (\cos\mu - \sin\mu)(x + y) - L_\omega^{-1}[-(1 - 2\sin^2\mu)(x - y)^2 + 4\mu\mathbf{e}], \tag{3.3}$$

$$\varphi'_x(z) = (\cos\mu + \sin\mu)I - (\cos\mu - \sin\mu)^2 L_\omega^{-1} L_{x-y}, \tag{3.4}$$

$$\varphi'_y(z) = (\cos\mu + \sin\mu)I + (\cos\mu - \sin\mu)^2 L_\omega^{-1} L_{x-y}, \tag{3.5}$$

and ω is defined by (2.4).

(ii) *If F is a continuously differentiable and monotone function, then $H'(z)$ is invertible for any $\mu \in (0, \pi/2)$.*

P r o o f. By Theorem 2.2, it is easy to show that (i) holds. Now we prove (ii). Fix any $\mu > 0$. Let $\tilde z := (\tilde\mu, \tilde x, \tilde y) \in \mathbb{R} \times \mathbb{R}^n \times \mathbb{R}^n$ be an arbitrary vector which satisfies

$H'(z)\tilde{z} = 0$. It is sufficient to prove that $\tilde{\mu} = 0$, $\tilde{x} = 0$, and $\tilde{y} = 0$. From (3.2), $H'(z)\tilde{z} = 0$ gives

$$\begin{aligned} \tilde{\mu} &= 0, \\ F'(x)\tilde{x} - \tilde{y} &= 0, \\ \varphi'_x(z)\tilde{x} + \varphi'_y(z)\tilde{y} &= 0. \end{aligned} \tag{3.6}$$

By the second equation in (3.6) and the monotonicity of F, we have

$$\langle \tilde{x}, \tilde{y} \rangle = \langle \tilde{x}, F'(x)\tilde{x} \rangle \geqslant 0. \tag{3.7}$$

By (3.4) and (3.5), the third equation in (3.6) yields

$$\begin{aligned} &[(\cos\mu + \sin\mu)I - (\cos\mu - \sin\mu)^2 L_\omega^{-1} L_{x-y}]\tilde{x} \\ &\qquad + [(\cos\mu + \sin\mu)I + (\cos\mu - \sin\mu)^2 L_\omega^{-1} L_{x-y}]\tilde{y} = 0. \end{aligned}$$

Since $L_x y = x \circ y$ for any $x, y \in \mathbb{R}^n$, the above equality is equivalent to

$$(\cos\mu + \sin\mu)(\tilde{x} + \tilde{y}) - (\cos\mu - \sin\mu)^2 L_\omega^{-1}[(x - y) \circ (\tilde{x} - \tilde{y})] = 0. \tag{3.8}$$

Premultiplying (3.8) by L_ω yields

$$(\cos\mu + \sin\mu)\omega \circ (\tilde{x} + \tilde{y}) - (\cos\mu - \sin\mu)^2[(x - y) \circ (\tilde{x} - \tilde{y})] = 0. \tag{3.9}$$

Since

$$\begin{aligned} (\cos\mu + \sin\mu)(\tilde{x} + \tilde{y}) &= (\cos\mu\tilde{x} + \sin\mu\tilde{y}) + (\sin\mu\tilde{x} + \cos\mu\tilde{y}), \\ (\cos\mu - \sin\mu)(\tilde{x} - \tilde{y}) &= (\cos\mu\tilde{x} + \sin\mu\tilde{y}) - (\sin\mu\tilde{x} + \cos\mu\tilde{y}), \end{aligned}$$

it follows from (3.9) that

$$\begin{aligned} &[\omega - (\cos\mu - \sin\mu)(x - y)] \circ [\cos\mu\tilde{x} + \sin\mu\tilde{y}] \\ &\qquad + [\omega + (\cos\mu - \sin\mu)(x - y)] \circ [\sin\mu\tilde{x} + \cos\mu\tilde{y}] = 0. \end{aligned} \tag{3.10}$$

Due to the definition of ω, we have

$$\omega^2 = (\cos\mu - \sin\mu)^2(x - y)^2 + 4\mu^2\mathbf{e} \succ_{\mathcal{K}} (\cos\mu - \sin\mu)^2(x - y)^2.$$

Then, by Proposition 3.4 in [16], we get

$$\omega - (\cos\mu - \sin\mu)(x - y) \succ_{\mathcal{K}} 0, \quad \omega + (\cos\mu - \sin\mu)(x - y) \succ_{\mathcal{K}} 0. \tag{3.11}$$

Note that

$$[\omega - (\cos\mu - \sin\mu)(x-y)] \circ [\omega + (\cos\mu - \sin\mu)(x-y)] = 4\mu^2\mathbf{e} \succ_{\mathcal{K}} 0, \tag{3.12}$$

$$\langle \cos\mu\tilde{x} + \sin\mu\tilde{y}, \sin\mu\tilde{x} + \cos\mu\tilde{y}\rangle \geqslant 0, \tag{3.13}$$

where the inequality (3.13) follows from (3.7). Set

$$a := \omega - (\cos\mu - \sin\mu)(x-y), \quad b := \omega + (\cos\mu - \sin\mu)(x-y),$$
$$u := \cos\mu\tilde{x} + \sin\mu\tilde{y}, \quad v := \sin\mu\tilde{x} + \cos\mu\tilde{y}.$$

Then, from (3.10)–(3.13) and using Lemma 3.1, we obtain that

$$\cos\mu\tilde{x} + \sin\mu\tilde{y} = 0, \quad \sin\mu\tilde{x} + \cos\mu\tilde{y} = 0,$$

which yields

$$\cos\mu\mathbf{e} \circ \tilde{x} + \sin\mu\mathbf{e} \circ \tilde{y} = 0, \quad \sin\mu\mathbf{e} \circ \tilde{x} + \cos\mu\mathbf{e} \circ \tilde{y} = 0. \tag{3.14}$$

Since $\cos\mu > 0$ and $\sin\mu > 0$ for any $\mu \in (0, \pi/2)$, we obtain that

$$\cos\mu\mathbf{e} \succ_{\mathcal{K}} 0, \quad \sin\mu\mathbf{e} \succ_{\mathcal{K}} 0, \quad \cos\mu\mathbf{e} \circ \sin\mu\mathbf{e} = \frac{\sin 2\mu}{2}\mathbf{e} \succ_{\mathcal{K}} 0.$$

Then, from (3.7) and (3.14), also using Lemma 3.1, we have $\tilde{x} = \tilde{y} = 0$. Thus, the null space of $H'(z)$ comprises only the origin, and hence $H'(z)$ is invertible. $\square$

For any $z := (\mu, x, y) \in \mathbb{R}_{++} \times \mathbb{R}^n \times \mathbb{R}^n$, we define the norm-function as follows:

$$\Psi(z) := \|H(z)\|^2. \tag{3.15}$$

Now we give our smoothing Newton method.

Algorithm 3.1 (A smoothing Newton method for the SOCCP).

Step 0: Choose an accuracy parameter $\varepsilon > 0$. Choose constants $\delta, \sigma \in (0,1)$, and $\mu_0 \in (0, \pi/2)$. Let $(x^0, y^0) \in \mathbb{R}^n \times \mathbb{R}^n$ be an arbitrary initial point. Set $\overline{z} := (\mu_0, 0, 0)$. Set $z^0 := (\mu_0, x^0, y^0)$. Take $\tau \in (0,1)$ such that $\mu_0\tau < 1/2$ and $\tau\|H(z^0)\| < 1$. Set $k := 0$.

Step 1: If $\|H(z^k)\| \leqslant \varepsilon$, then stop. Otherwise, compute

$$\beta_k := \beta(z^k) = \tau \min\{1, \|H(z^k)\|\}. \tag{3.16}$$

Step 2: Compute $\Delta z^k := (\Delta\mu_k, \Delta x^k, \Delta y^k) \in \mathbb{R} \times \mathbb{R}^n \times \mathbb{R}^n$ by

$$H'(z^k)\Delta z^k = -H(z^k) + \beta_k\|H(z^k)\|\overline{z}, \tag{3.17}$$

where $H'(z^k)$ denotes the Jacobian of the function H at the point z^k.

Step 3: Choose $\alpha_k := \delta^{l_k}$, where l_k is the first nonnegative integer l for which

$$\Psi(z^k + \delta^l \Delta z^k) \leqslant [1 - \sigma(1 - 2\mu_0\tau)\delta^l]\Psi(z^k), \tag{3.18}$$

i.e., $l = 0, 1, \ldots$ are tried successively until the above inequality is satisfied for $l = l_k$.

Step 4: Set $z^{k+1} := z^k + \alpha_k \Delta z^k$ and $k := k + 1$. Go to Step 1.

We note that Algorithm 3.1 solves only one system of linear equations and performs only one Armijo-type line search at each iteration. If $\|H(z^k)\| = 0$, then (x^k, y^k) is the solution to the SOCCP. So, the stopping criterion is reasonable. In the following, we give some remarks on Algorithm 3.1.

Remark 3.1. Suppose that F is a continuously differentiable and monotone function and that $\{z^k\}$ is the iteration sequence generated by Algorithm 3.1. Then, the following results hold:

(a) From (3.18), it is easy to see that the sequence $\{\Psi(z^k)\}$ is monotonically decreasing, and hence, sequences $\{\|H(z^k)\|\}$ and $\{\beta_k\}$ are monotonically decreasing.

(b) Denote

$$\Omega := \{z := (\mu, x, y) \in \mathbb{R}_{++} \times \mathbb{R}^n \times \mathbb{R}^n \colon\ \mu \geqslant \mu_0\beta(z)\|H(z)\|\}, \tag{3.19}$$

where $\beta(\cdot)$ and μ_0 are given in (3.16) and Step 0 of Algorithm 3.1, respectively. Then, $z^k \in \Omega$ for all $k \geqslant 0$. This can be obtained by induction. Since

$$\beta(z^0)\|H(z^0)\| = \tau\|H(z^0)\| \min\{1, \|H(z^0)\|\} \leqslant \tau\|H(z^0)\| < 1,$$

we have $\mu_0 \geqslant \mu_0\beta(z^0)\|H(z^0)\|$. So, $z^0 \in \Omega$. Suppose that $z^k \in \Omega$, i.e., $\mu_k \geqslant \mu_0\beta_k\|H(z^k)\|$. Then

$$\begin{aligned}
\mu_{k+1} &- \mu_0\beta_{k+1}\|H(z^{k+1})\| \\
&= \mu_k + \alpha_k\Delta\mu_k - \mu_0\beta_{k+1}\|H(z^{k+1})\| \\
&= (1 - \alpha_k)\mu_k + \alpha_k\mu_0\beta_k\|H(z^k)\| - \mu_0\beta_{k+1}\|H(z^{k+1})\| \\
&\geqslant (1 - \alpha_k)\mu_0\beta_k\|H(z^k)\| + \alpha_k\mu_0\beta_k\|H(z^k)\| - \mu_0\beta_{k+1}\|H(z^{k+1})\| \\
&= \mu_0(\beta_k\|H(z^k)\| - \beta_{k+1}\|H(z^{k+1})\|) \geqslant 0,
\end{aligned}$$

where the second equality follows from the first equation in (3.17); the first inequality from the assumption that $z^k \in \Omega$; and the last inequality from the result (a). That is, $\mu_{k+1} \geqslant \mu_0 \beta_{k+1} \|H(z^{k+1})\|$. Thus, $z^k \in \Omega$ for all $k \geqslant 0$.

(c) $0 < \mu_k < \pi/2$ for all $k \geqslant 0$. First, we prove $\mu_k > 0$ for all $k \geqslant 0$ by induction on k. Suppose that $\mu_k > 0$ for some k, e.g., it is satisfied for $k = 0$. Then, from the first equation in (3.17), we get for any $\alpha \in (0, 1)$

$$\mu_k + \alpha \Delta \mu_k = \mu_k + \alpha(-\mu_k + \mu_0 \beta_k \|H(z^k)\|) = (1-\alpha)\mu_k + \alpha \mu_0 \beta_k \|H(z^k)\| > 0,$$

which implies that $\mu_{k+1} > 0$. Thus, we have $\mu_k > 0$ for all $k \geqslant 0$. On the other hand, from the result (b), we obtain that for all $k \geqslant 0$

$$\mu_{k+1} = (1-\alpha_k)\mu_k + \alpha_k \mu_0 \beta_k \|H(z^k)\| \leqslant (1-\alpha_k)\mu_k + \alpha_k \mu_k = \mu_k.$$

Hence, the sequence $\{\mu_k\}$ is monotonically decreasing. This shows that $\mu_k \leqslant \mu_0 < \pi/2$ holds for all $k \geqslant 0$. By Theorem 3.2, we know that $H'(z^k)$ is nonsingular for all $k \geqslant 0$. Thus, the system of equations (3.17) is solvable, i.e., Step 2 is well-defined at the kth iteration.

(d) For any $\alpha \in (0, 1)$, denote

$$F_k(\alpha) = \Psi(z^k + \alpha \Delta z^k) - \Psi(z^k) - \alpha \Psi'(z^k)^{\mathrm{T}} \Delta z^k. \tag{3.20}$$

We obtain that $F_k(\alpha) = o(\alpha)$, since $\Psi(\cdot)$ is continuously differentiable around z^k. Then, from (3.17) and (3.20) we have

$$\begin{aligned}
\Psi(z^k + \alpha \Delta z^k) &= \Psi(z^k) + \alpha \Psi'(z^k)^{\mathrm{T}} \Delta z^k + F_k(\alpha) \\
&= \Psi(z^k) + 2\alpha H(z^k)^{\mathrm{T}} H'(z^k) \Delta z^k + o(\alpha) \\
&= \Psi(z^k) + 2\alpha H(z^k)^{\mathrm{T}} [-H(z^k) + \beta_k \|H(z^k)\| \overline{z}] + o(\alpha) \\
&= (1-2\alpha)\Psi(z^k) + 2\alpha \beta_k \|H(z^k)\| H(z^k)^{\mathrm{T}} \overline{z} + o(\alpha) \\
&\leqslant (1-\alpha)\Psi(z^k) + 2\alpha \mu_0 \tau \Psi(z^k) + o(\alpha) \\
&= [1 - (1 - 2\mu_0 \tau)\alpha] \Psi(z^k) + o(\alpha),
\end{aligned}$$

where the inequality follows from the fact that $\|\overline{z}\| = \mu_0$ and $\beta_k \leqslant \tau$ by Step 0 and Step 1 of Algorithm 3.1. Since $\mu_0 \tau < 1/2$, there exists a constant $\overline{\alpha} \in (0, 1)$ such that

$$\Psi(z^k + \alpha \Delta z^k) \leqslant [1 - \sigma(1 - 2\mu_0 \tau)\alpha] \Psi(z^k)$$

holds for any $\alpha \in (0, \overline{\alpha})$ and $\sigma \in (0, 1)$. This demonstrates that Step 3 is well-defined at the kth iteration.

4. Convergence analysis

In this section, we show that any accumulation point z^* of the iteration sequence $\{z^k\}$ is a solution to the system $H(z) = 0$. If the accumulation point z^* satisfies a nonsingularity assumption, then the iteration sequence $\{z^k\}$ converges to z^* locally quadratically without strict complementarity. First, we prove the global convergence of Algorithm 3.1. For this purpose, we need the following result:

Lemma 4.1. *Let F be a continuously differentiable and monotone function and $\{z^k\}$ be the iteration sequence generated by Algorithm 3.1. If there exists $\mu^* > 0$ such that $\mu_k > \mu^*$ holds for all $k \geqslant 0$ and $\|(x^k, y^k)\| \to \infty$ $(k \to \infty)$, then*

$$\lim_{k\to\infty} \|H(z^k)\| = \infty.$$

Proof. By assuming that $\{\|H(z^k)\|\}$ is bounded, we will derive a contradiction. It follows from (3.1) that

$$\|H(z^k)\|^2 = \mu_k^2 + \|F(x^k) - y^k\|^2 + \|\varphi(\mu_k, x^k, y^k)\|^2. \tag{4.1}$$

Then, we obtain that $\{F(x^k) - y^k\}$ and $\{\varphi(\mu_k, x^k, y^k)\}$ are bounded. Denote

$$\beta(x^k, y^k) := y^k - F(x^k).$$

It is easy to see that $\{\beta(x^k, y^k)\}$ is bounded and $y^k = \beta(x^k, y^k) + F(x^k)$. Suppose that $\{u^k\}$ is an arbitrary bounded sequence. Then $\{F(u^k)\}$ is bounded by the continuity of F. Let $v^k := \beta(x^k, y^k) + F(u^k)$. Obviously, $\{v^k\}$ is bounded. Since F is monotone, it follows from Definition 1.2 that

$$\langle x^k - u^k, y^k - v^k\rangle = \langle x^k - u^k, F(x^k) - F(u^k)\rangle \geqslant 0.$$

Moreover, $\mu_k \in (\mu^*, \pi/2)$. Thus, from Theorem 2.3 we have

$$\lim_{\|(x^k, y^k)\|\to\infty} \|\varphi(\mu_k, x^k, y^k)\| = \infty,$$

which, together with (4.1), shows that $\{\|H(z^k)\|\}$ is unbounded. A contradiction is derived. The proof is completed. □

Now we are in the position to give the global convergence of Algorithm 3.1. To this end, we need the following assumption:

Assumption 4.1. The solution set of the SOCCP defined by $S := \{(x, y) \in \mathbb{R}^n \times \mathbb{R}^n \colon x \in \mathcal{K},\ y \in \mathcal{K},\ x^{\mathrm{T}} y = 0,\ y = F(x)\}$ is nonempty and bounded.

Theorem 4.2 (Global convergence). *Let F be a continuously differentiable and monotone function. Suppose that $\{z^k\}$ is the iteration sequence generated by Algorithm* 3.1. *Then the following results hold:*

(i) $\{\|H(z^k)\|\}$ *is monotonically decreasing and converges to zero.*

(ii) $\lim\limits_{k\to\infty} \mu_k = 0$. *Moreover, if Assumption* 4.1 *holds, then $\{z^k\}$ is bounded.*

(iii) *Any accumulation point of $\{z^k\}$ is a solution to* (1.1).

P r o o f. According to the result (a) in Remark 3.1, we know that $\{\|H(z^k)\|\}$ is monotonically decreasing and bounded from below by zero. Thus, there exists $H^* \geqslant 0$ and $\beta^* \geqslant 0$ such that $\lim\limits_{k\to\infty} \|H(z^k)\| = H^*$ and $\lim\limits_{k\to\infty} \beta_k = \beta^*$. If $H^* = 0$, then we obtain the desired result. Suppose that $H^* > 0$. Then from (3.16) we have $\beta^* > 0$. Since $z^k \in \Omega$ for all $k \geqslant 0$ by the result (b) in Remark 3.1, we obtain that $\mu_k \geqslant \mu_0\beta^* H^* > 0$. Therefore, from Lemma 4.1, we know that $\{z^k := (\mu_k, x^k, y^k)\}$ is bounded and hence it has at least one accumulation point $z^* := (\mu^*, x^*, y^*)$. Without loss of generality, we assume that $\{z^k\}$ converges to z^* as $k \to \infty$. Since $z^* \in \Omega$, we have $0 < \mu_0\beta^* H^* \leqslant \mu^* < \pi/2$ by the results (b) and (c) in Remark 3.1. It follows from Theorem 3.2 that $H'(z^*)$ exists and is invertible. Hence, there exists a closed neighborhood $N(z^*)$ of z^* such that for any $z \in N(z^*)$ we have $0 < \mu < \pi/2$ and $H'(z)$ is invertible. Then, for any $z \in N(z^*)$, let $\Delta z := (\Delta\mu, \Delta x, \Delta y) \in \mathbb{R} \times \mathbb{R}^n \times \mathbb{R}^n$ be the unique solution to the system of equations

$$H'(z)\Delta z = -H(z) + \beta(z)\|H(z)\|\overline{z}.$$

Denote

$$g_z(\alpha) := \Psi(z + \alpha\Delta z) - \Psi(z) - \alpha\Psi'(z)^{\mathrm{T}}\Delta z.$$

Then, for any $z \in N(z^*)$, we have $\lim\limits_{\alpha\to 0} |g_z(\alpha)|/\alpha = 0$. Similarly to the result (d) in Remark 3.1, for any $\alpha \in (0,1)$ and $z \in N(z^*)$, we have

$$\Psi(z + \alpha\Delta z) \leqslant [1 - (1 - 2\mu_0\tau)\alpha]\Psi(z) + o(\alpha).$$

Hence, we can find a positive number $\overline{\alpha} \in (0,1]$ such that

$$\Psi(z + \alpha\Delta z) \leqslant [1 - \sigma(1 - 2\mu_0\tau)\alpha]\Psi(z)$$

holds for any $\alpha \in (0, \overline{\alpha}]$, $\sigma \in (0,1)$, and $z \in N(z^*)$. Therefore, for all sufficiently large k, there exists a nonnegative integer $\overline{l}$ such that $\delta^{\overline{l}} \in (0, \overline{\alpha}]$ and

$$\Psi(z^k + \delta^{\overline{l}}\Delta z^k) \leqslant [1 - \sigma(1 - 2\mu_0\tau)\delta^{\overline{l}}]\Psi(z^k).$$

For all sufficiently large k, since $\alpha_k = \delta^{l_k} \geqslant \delta^{\bar{l}}$, it follows from Step 3 and Step 4 in Algorithm 3.1 that

$$\Psi(z^{k+1}) \leqslant [1 - \sigma(1 - 2\mu_0\tau)\delta^{l_k}]\Psi(z^k) \leqslant [1 - \sigma(1 - 2\mu_0\tau)\delta^{\bar{l}}]\Psi(z^k),$$

which implies that $\Psi(z^{k+1}) \leqslant C\Psi(z^k)$, where $C = 1-\sigma(1-2\mu_0\tau)\delta^{\bar{l}} < 1$ is a constant, and thus $\Psi(z^k) \to 0$ as $k \to \infty$, i.e., $\|H(z^k)\| \to 0$ as $k \to \infty$. This contradicts the fact that the sequence $\{\|H(z^k)\|\}$ converges to $H^* > 0$. Thus, $\{\|H(z^k)\|\}$ is monotonically decreasing and converges to zero.

Next, we prove (ii). Due to the definition of H and $\lim\limits_{k\to\infty} \|H(z^k)\| = 0$, it is easy to see that $\lim\limits_{k\to\infty} \mu_k = 0$. From the result (b) in Remark 3.1, we know that $z^k \in \Omega$ for all $k \geqslant 0$. This gives

$$\|H(z^k)\| \leqslant \frac{\mu_k}{\mu_0\beta(z^k)}.$$

Denote

$$\Phi(z^k) := \begin{pmatrix} F(x^k) - y^k \\ \varphi(z^k) \end{pmatrix}, \quad \lambda_k := \max\left\{\frac{\mu_k}{\mu_0\tau}, \sqrt{\frac{\mu_k}{\mu_0\tau}}\right\}.$$

Since $\beta(z^k) = \tau$ or $\beta(z^k) = \tau\|H(z^k)\|$, we have $\|\Phi(z^k)\| \leqslant \|H(z^k)\| \leqslant \lambda_k$ and $\lim\limits_{k\to\infty} \lambda_k = 0$. Hence, under Assumption 4.1, by using the famous mountain pass theorem (e.g., Theorem 3.5 in [32]), we can prove that $\{(x^k, y^k)\}$ is bounded similarly as Theorem 3.1 in [18]. Therefore, $\{z^k\}$ is bounded and hence it has at least one accumulation point $z^* := (\mu^*, x^*, y^*)$.

Finally, we prove (iii). Let z^* be an accumulation point of $\{z^k\}$. Then, there exists a subsequence, which we write without loss of generality as $\{z^k\}$, such that $\{z^k\}$ converges to z^* as $k \to \infty$. It follows from the continuity of H that $\lim\limits_{k\to\infty} \|H(z^k)\| = \|H(z^*)\|$. Thus, from the result (i) we know that $H(z^*) = 0$ and z^* is a solution to (1.1). So, we complete the proof. $\square$

Next, we discuss the local convergence of Algorithm 3.1. For this purpose, we need the concept of semismoothness which was originally introduced by Mifflin [21] for functionals. Qi and Sun [25] extended the definition of semismooth function to vector-valued functions. A locally Lipschitz function $F\colon \mathbb{R}^n \to \mathbb{R}^m$, which has the generalized Jacobian $\partial F(z)$ as in Clarke [13], is said to be semismooth at $x \in \mathbb{R}^n$ if $\lim\limits_{V\in\partial F(x+th'), h'\to h, t\downarrow 0} \{Vh'\}$ exists for any $h \in \mathbb{R}^n$; F is said to be strongly semismooth at x if F is semismooth at x and, for any $V \in \partial F(x + h)$, $h \to 0$, it follows that

$$F(x + h) - F(x) - Vh = O(\|h\|^2). \tag{4.2}$$

Lemma 4.3. *Let* $\varphi(\mu, x, y)$ *be defined by* (2.3). *Then,* $\varphi(\mu, x, y)$ *is strongly semismooth at all points* $(\mu, x, y) \in \mathbb{R}_{++} \times \mathbb{R}^n \times \mathbb{R}^n$.

Proof. Theorem 4.2 in [9] shows that the function $g(x, \mu) = \sqrt{x^2 + \mu^2 \mathbf{e}}$ is strongly semismooth at $(x, \mu) \in \mathbb{R}^n \times \mathbb{R}$. By recalling the definition of φ and the fact that the composition of strongly semismooth functions is strongly semismooth, we can obtain that $\varphi(\mu, x, y)$ is strongly semismooth at all points $(\mu, x, y) \in \mathbb{R}_{++} \times \mathbb{R}^n \times \mathbb{R}^n$. □

Since a vector-valued function is (strongly) semismooth if and only if its all component functions are (strongly) semismooth, by Lemma 4.3, we obtain the following corollary:

Corollary 4.4. *Let* $H(z)$ *be defined by* (3.1). *Then, there exists a neighborhood* $N(z^*)$ *of* z^* *such that* $H(z)$ *is semismooth at any point* $z \in N(z^*)$. *Furthermore,* $H(z)$ *is strongly semismooth at any point* $z \in N(z^*)$ *if* $F'(x)$ *is locally Lipschitz continuous around* x^*.

To obtain the local quadratic convergence of the algorithm, we need a few conditions. For related smoothing methods, a typical condition is that z^* satisfies the strict complementarity condition and is nondegenerate (e.g., [3]–[5]). In this paper, we use the following assumption:

Assumption 4.2.
(i) All $V \in \partial H(z^*)$ are nonsingular.
(ii) $F'(x)$ is locally Lipschitz continuous around x^*.

Theorem 4.5. *Suppose that* F *is a continuously differentiable and monotone function and that* z^* *is an accumulation point of the iteration sequence* $\{z^k\}$ *generated by Algorithm* 3.1. *If Assumption* 4.2 *holds, then*
(a) $z^{k+1} = z^k + \Delta z^k$ *for all sufficiently large* k,
(b) $\{z^k\}$ *converges to* z^* *quadratically, i.e.,* $\|z^{k+1} - z^*\| = O(\|z^k - z^*\|^2)$; *moreover,* $\mu_{k+1} = O(\mu_k^2)$.

Proof. Theorem 4.2 shows that $H(z^*) = 0$. Since all $V \in \partial H(z^*)$ are nonsingular, it follows from Proposition 3.1 in [25] that $\|H'(z^k)^{-1}\| = O(1)$ holds for all z^k sufficiently close to z^*. Since $H(\cdot)$ is strongly semismooth around z^*, for all z^k sufficiently close to z^*, we have

$$\|H(z^k) - H(z^*) - H'(z^k)(z^k - z^*)\| = O(\|z^k - z^*\|^2). \tag{4.3}$$

Note that H is locally Lipschitz continuous near z^* since it is semismooth. Thus, $\|H(z^k)\| = O(\|z^k - z^*\|)$ holds for all z^k sufficiently close to z^*. So, for all z^k sufficiently close to z^*, it follows from (3.17) and (4.3) that

$$
\begin{aligned}
(4.4)\quad \|z_k + \Delta z^k - z^*\| & \\
&= \|z_k + H'(z^k)^{-1}[-H(z^k) + \beta_k\|H(z^k)\|\overline{z}] - z^*\| \\
&\leqslant \|H'(z^k)^{-1}\|[\|H(z^k) - H(z^*) - H'(z^k)(z^k - z^*)\| + \mu_0\tau\|H(z^k)\|^2] \\
&= O(\|z^k - z^*\|^2),
\end{aligned}
$$

where the inequality uses the fact that $\beta_k = \tau\|H(z^k)\|$ for all z^k sufficiently close to z^*. Similarly as the proof of Theorem 3.1 in [24], we have $\|z^k - z^*\| = O(\|H(z^k)\|)$ for all z^k sufficiently close to z^*. Hence, it follows from (4.4) that for all z^k sufficiently close to z^*,

$$
(4.5)\qquad \|H(z^k + \Delta z^k)\| = O(\|z^k + \Delta z^k - z^*\|) = O(\|z^k - z^*\|^2) = O(\|H(z^k)\|^2).
$$

From Theorem 4.2, we know that $\|H(z^k)\| \to 0$ as $k \to \infty$. Thus, (4.5) implies that $\alpha_k = 1$ for all z^k sufficiently close to z^*. This, together with (4.4), indicates that the result (a) holds, and for all z^k sufficiently close to z^*,

$$
\|z^{k+1} - z^*\| = O(\|z^k - z^*\|^2).
$$

Next, we prove the second part of (b). It follows from (a) that for all sufficiently large k,

$$
(4.6)\qquad \mu_{k+1} = \mu_k + \Delta\mu_k = \mu_0\beta_k\|H(z^k)\| = \mu_0\tau\|H(z^k)\|^2.
$$

Thus, for all sufficiently large k, by using (4.5) and (4.6) we have

$$
\frac{\mu_{k+1}}{\mu_k^2} = \frac{\|H(z^k)\|^2}{\mu_0\tau\|H(z^{k-1})\|^4} = \frac{O(\|H(z^{k-1})\|^4)}{\mu_0\tau\|H(z^{k-1})\|^4},
$$

which implies that $\mu_{k+1} = O(\mu_k^2)$. This completes the proof. $\square$

5. Numerical experiments

In this section, we conduct some numerical experiments to solve some SOCCP's by Algorithm 3.1 and report the computational results. All experiments were performed on a personal computer with 2.0 GB memory and Intel(R) Pentium(R) Dual-Core CPU 2.93 GHz×2. The operating system was Windows XP and the computer codes were all written in Matlab 7.0.1. In all the experiments, we used $\|H(z^k)\| \leqslant 10^{-8}$ as the stopping criterion.

5.1. Linear case

Find $(x, y) \in \mathbb{R}^n \times \mathbb{R}^n$ such that

$$x \in \mathcal{K}, \quad y \in \mathcal{K}, \quad x^{\mathrm{T}} y = 0, \quad y = Mx + q, \tag{5.1}$$

where $M \in \mathbb{R}^{n\times n}$ and $q \in \mathbb{R}^n$. We used $x^0 = \mathbf{e}$ and $y^0 = 0$ as the starting points.

In the first set of experiments, we set

$$M = \operatorname{diag}(1/n, 2/n, \ldots, 1), \quad q = (-1, -1, \ldots, -1)^{\mathrm{T}}.$$

The parameters used in Algorithm 3.1 were chosen as $\mu_0 = 0.1$, $\sigma = 0.5$, $\delta = 0.8$, $\tau = 0.95/(1 + \|H(z^0)\|)$. The results are listed in Tab. 1, where n denotes the problem size; **IT** and **CPU** denote the number of iterations and the CPU time in seconds, respectively; **FV** and **MU** denote the value of $\|H(z^k)\|$ and μ_k when the algorithm terminates.

n	**IT**	**CPU**	**FV**	**MU**
8	6	0.015	1.6874×10^{-9}	1.4683×10^{-9}
16	8	0.015	4.2556×10^{-11}	2.8190×10^{-14}
32	9	0.016	4.8565×10^{-10}	6.5322×10^{-14}
64	11	0.016	2.5334×10^{-9}	2.3882×10^{-10}
128	15	0.531	6.1484×10^{-14}	1.5337×10^{-18}
256	21	5.516	4.1897×10^{-14}	2.9229×10^{-17}

Table 1. Numerical results for the linear SOCCP of various problem sizes.

In the second set of experiments, the matrix M was obtained by setting $M = N^{\mathrm{T}}N$, where N was a square matrix. Elements of N and q were chosen randomly from the interval $[0, 1]$. The random problems of each case were generated 10 times. Tab. 2 gives the results when we chose parameters in Algorithm 3.1 as $\mu_0 = 0.1$, $\sigma = 0.5$, $\delta = 0.8$, $\tau = 0.95/(1 + \|H(z^0)\|)$, in which n denotes the problem size;

MIT and **MCPU** denote the maximum values of the number of iterations and the CPU time in seconds, respectively; **AIT** and **ACPU** denote the average values of the number of iterations and the CPU time in seconds, respectively; **MFV** and **MGAP** denote the maximum values of $\|H(z)\|$ and $|x^{\mathrm{T}}y|$ at the final iteration in the 10 trials, respectively.

n	**MIT**	**AIT**	**MCPU**	**ACPU**	**MFV**	**MGAP**
100	7	6.4	0.15	0.12	8.0853×10^{-9}	3.9338×10^{-9}
200	9	7.3	1.05	0.85	2.0309×10^{-10}	5.2779×10^{-11}
300	8	7.8	3.00	2.92	2.3614×10^{-9}	9.5040×10^{-10}
400	9	8.5	7.63	7.16	7.6918×10^{-9}	2.6244×10^{-9}
500	10	8.8	16.03	14.11	4.6039×10^{-9}	1.1348×10^{-9}
600	9	8.6	24.38	23.28	7.4057×10^{-10}	2.2742×10^{-10}
700	9	8.8	38.19	37.26	7.5673×10^{-10}	1.9741×10^{-10}
800	12	9.4	73.31	55.65	7.8235×10^{-9}	1.5626×10^{-9}

Table 2. Numerical results for the linear SOCCP of various problem sizes.

Table 3 gives the results for our second set of experiments when we chose parameters in Algorithm 3.1 as $\mu_0 = 0.1$, $\sigma = 0.25$, $\delta = 0.5$, $\tau = 0.5/(2 + \|H(z^0)\|)$.

n	**MIT**	**AIT**	**MCPU**	**ACPU**
100	7	6.4	0.14	0.12
200	8	7.2	0.94	0.84
300	8	7.3	3.00	2.68
400	9	7.9	7.65	6.71
500	9	8.2	14.59	13.26
600	9	8.1	24.56	22.11
700	10	8.5	42.77	36.34
800	12	9.2	75.50	54.89

Table 3. Numerical results for the linear SOCCP of various problem sizes.

From the results in Tabs. 2 and 3, we may see that Algorithm 3.1 is effective for solving the SOCCP (5.1). It can solve all the test problems and can deal with large-scale SOCCP problems. It can find a solution point meeting the desired accuracy in few iterations and in short CPU time. In addition, we also find that there are slight changes in results for different values of σ, δ, and τ.

In our third set of experiments, elements of q were chosen randomly from the interval $[-1, 1]$ and $M = N^{\mathrm{T}}N$, where $N \in \mathbb{R}^{n\times n}$ was a sparse matrix (the density of N is nonzero) whose elements were chosen randomly from the interval $[0, 1]$. The random problems were generated 10 times for each nonzero density with problem size $n = 100$ and $n = 500$. The parameters used in Algorithm 3.1 were chosen as $\mu_0 = 0.1$, $\sigma = 0.5$, $\delta = 0.8$, $\tau = 0.95/(1 + \|H(z^0)\|)$. The tested results are listed in Tabs. 4 and 5, where **Dens.** denotes the nonzero density of matrix N; **AIT** and **ACPU** denote the average values of the number of iterations and the CPU time in seconds, respectively; **MFV** and **MMU** denote the maximum values of $\|H(z)\|$ and μ at the final iteration in the 10 trials for each nonzero density, respectively.

Dens. (%)	**AIT**	**ACPU**	**MFV**	**MMU**
5	7.6	0.217	6.0586×10^{-10}	3.5705×10^{-12}
10	7.8	0.245	1.5797×10^{-9}	2.2005×10^{-11}
20	7.1	0.224	5.2156×10^{-9}	7.3635×10^{-11}
40	7.2	0.241	7.6694×10^{-9}	4.6839×10^{-11}
60	7.2	0.209	5.7693×10^{-9}	1.9911×10^{-11}
80	7.3	0.236	3.0560×10^{-9}	3.7419×10^{-12}

Table 4. Numerical results for the linear SOCCP with different nonzero density ($n = 100$).

Dens. (%)	**AIT**	**ACPU**	**MFV**	**MMU**
5	7.9	23.21	1.0290×10^{-9}	2.6611×10^{-11}
15	8.0	23.87	3.9807×10^{-9}	7.5880×10^{-12}
35	8.2	23.18	8.5395×10^{-9}	4.1456×10^{-12}
55	8.2	22.37	9.2832×10^{-9}	1.5064×10^{-12}
75	7.8	20.47	8.7656×10^{-10}	1.3906×10^{-13}
95	8.3	22.49	2.4070×10^{-9}	1.9853×10^{-13}

Table 5. Numerical results for the linear SOCCP with different nonzero density ($n = 500$).

The results in Tabs. 4 and 5 show that our algorithm performs well. It has good convergence and numerical stability. The number of iterations and CPU time slightly change with the sparsity of N. We have also tested some other problems, and the computation effect is similar.

5.2. Nonlinear case

Find $(x, y) \in \mathbb{R}^5 \times \mathbb{R}^5$ such that

$$x \in \mathcal{K}, \quad y \in \mathcal{K}, \quad x^{\mathrm{T}} y = 0, \quad y = F(x), \tag{5.2}$$

where $\mathcal{K} = \mathcal{K}^3 \times \mathcal{K}^2$ and $F\colon \mathbb{R}^5 \to \mathbb{R}^5$ is given by

$$F(x) = \begin{pmatrix} 24(2x_1 - x_2)^3 + \exp(x_1 - x_3) - 4x_4 + x_5 \\ -12(2x_1 - x_2)^3 + 3(3x_2 + 5x_3)/\sqrt{1 + (3x_2 + 5x_3)^2} - 6x_4 - 7x_5 \\ -\exp(x_1 - x_3) + 5(3x_2 + 5x_3)/\sqrt{1 + (3x_2 + 5x_3)^2} - 3x_4 + 5x_5 \\ 4x_1 + 6x_2 + 3x_3 - 1 \\ -x_1 + 7x_2 - 5x_3 + 2 \end{pmatrix}.$$

From the analysis in [17], we know that the function $F(x)$ is monotone.

In the experiments, the initial points of Algorithm 3.1 were chosen randomly. We set the parameters in Algorithm 3.1 as $\sigma = 0.5$, $\delta = 0.8$, $\tau = 0.95/(1 + \|H(z^0)\|)$ and μ_0 was a random number in $[1, 10]$. We tested 10 times for this problem. Numerical results are summarized in Tab. 6, where **IT** and **CPU** denote the number of iterations and the CPU time in seconds, respectively; **FV** and **GAP** denote the values of $\|H(z)\|$ and $|x^{\mathrm{T}}y|$ at the final iteration, respectively.

μ_0	**IT**	**CPU**	**FV**	**GAP**
7.3927	12	0.017	2.1610×10^{-12}	4.1943×10^{-13}
8.2521	20	0.016	3.0839×10^{-11}	3.3006×10^{-12}
1.7885	12	0.015	1.5593×10^{-15}	1.3878×10^{-17}
3.7678	10	0.016	1.2750×10^{-10}	2.6257×10^{-11}
7.4862	11	0.015	3.1223×10^{-12}	5.8269×10^{-13}
4.8751	13	0.016	1.8743×10^{-15}	1.3878×10^{-16}
7.3947	11	0.017	4.7698×10^{-9}	8.9623×10^{-10}
2.7836	12	0.016	3.5686×10^{-14}	6.0091×10^{-15}
7.1534	12	0.015	1.6149×10^{-12}	2.7273×10^{-13}
6.6358	20	0.015	9.8931×10^{-10}	1.7717×10^{-10}

Table 6. Numerical results for the nonlinear SOCCP with various initial points.

5.3. Comparison with interior point method

We consider the following problem:

$$\text{(SOCP)} \qquad \begin{aligned} &\min \sum_{i=1}^{r} c_i^{\mathrm{T}} x_i \\ &\text{s.t.} \ \sum_{i=1}^{r} A_i x_i = b, \\ &\qquad x_i \in \mathcal{K}^{n_i}, \quad i = 1, \dots, r, \end{aligned}$$

where $c_i, x_i \in \mathbb{R}^{n_i}$, $A_i \in \mathbb{R}^{m \times n_i}$, and $\mathcal{K}^{n_i}$ is an n_i-dimensional second-order cone. Let $y \in \mathbb{R}^m$ be the dual variable of the equality constraint. The KKT system of the SOCP is

$$\sum_{i=1}^{r} A_i x_i = b, \quad x_i^{\mathrm{T}}(c_i - A_i^{\mathrm{T}} y) = 0, \quad x_i \in \mathcal{K}^{n_i}, \quad c_i - A_i^{\mathrm{T}} y \in \mathcal{K}^{n_i}, \ i = 1, \dots, r. \tag{5.3}$$

Let $n = \sum_{i=1}^{r} n_i$, $c = (c_1, \dots, c_r)$, $x = (x_1, \dots, x_r)$, $A = (A_1, \dots, A_r)$. We solved the problem (5.3) by Algorithm 3.1 and SDPT3 [29], a successful interior point method software for the SOCP. The tested problems were randomly generated with sizes $n(= 2m)$ from 100 to 400 with each $n_i = 5$. We generated a random matrix A and a random vector x in the second-order cone which give a right-hand side $b = Ax$ and hence the problem is feasible. Moreover, we generated a random vector c in the second-order cone, so the optimal value of the problem is obtainable. Throughout the computational experiments, the parameters used in Algorithm 3.1 were chosen as $\mu_0 = 2 \times 10^{-3}$, $\delta = 0.65$, $\sigma = 0.05$, $\tau = 0.95/(1 + \|H(z^0)\|)$. Denote $\mathbf{e}^{n_i} = (1, 0, \dots, 0)^{\mathrm{T}}$, an n_i-dimensional vector; and $\mathbf{e} = (\mathbf{e}^{n_1}, \dots, \mathbf{e}^{n_r})$. We chose $x^0 = \mathbf{e}$, $y^0 = 0$ as the starting points.

The random problems of each case were generated 5 times. Tab. 7 and Tab. 8 give the numerical results when we implemented Algorithm 3.1 and SDPT3 for the problem (5.3), respectively, where **mIT** denotes the minimum value of the number of iterations, **AIT** denotes the average value of the number of iterations, **MCPU** denotes the maximum value of the CPU time in seconds, **ACPU** denotes the average value of the CPU time in seconds, **AFV** denotes the average value of $\|H(z^k)\|$ when the algorithm terminates among the 5 tests, and **mFV** denotes the minimum value of $\|H(z^k)\|$ when the algorithm terminates among the 5 tests. Comparing Tab. 7 and Tab. 8, we may see that the numerical results of Algorithm 3.1 are better than those by SDPT3, either in terms of the number of iterations or the CPU time.

m	n	**mIT**	**AIT**	**MCPU**	**ACPU**	**AFV**	**mFV**
50	100	11	12.4	0.92	0.75	2.43×10^{-9}	5.72×10^{-10}
100	200	15	16.6	1.26	1.02	1.18×10^{-10}	2.78×10^{-12}
150	300	13	15.8	2.05	2.33	9.53×10^{-9}	6.57×10^{-11}
200	400	12	13.2	4.51	3.94	5.82×10^{-9}	3.44×10^{-10}

Table 7. Numerical results of Algorithm 3.1 for the problem (5.3).

m	n	**mIT**	**AIT**	**MCPU**	**ACPU**	**AFV**	**mFV**
50	100	13	14.2	1.21	0.81	5.94×10^{-9}	6.38×10^{-10}
100	200	15	18.4	3.77	1.65	4.78×10^{-9}	3.94×10^{-11}
150	300	14	17.2	5.18	3.63	1.71×10^{-9}	6.79×10^{-11}
200	400	16	19.6	5.96	4.88	1.32×10^{-10}	3.02×10^{-10}

Table 8. Numerical results of SDPT3 for the problem (5.3).

6. Conclusions

It has been shown in [33] that the Qi-Sun-Zhou smoothing Newton method [27] performs very efficiently for solving complementarity problems in practice. In this paper, by modifying and extending the Qi-Sun-Zhou method [27], we propose a smoothing Newton method for solving the SOCCP based on a new smoothing function. This new function is coercive under suitable conditions, which plays an important role in the convergence analysis. We prove that the proposed algorithm is globally and locally quadratically convergent under suitable assumptions. Some numerical results are also reported which indicate that our algorithm is effective for solving the SOCCP.

Acknowledgements. The authors sincerely thank the referees for their valuable suggestions on the paper which have greatly improved the paper.

References

[1] *J. Burke, S. Xu*: A non-interior predictor-corrector path-following algorithm for LCP. Reformulation: Nonsmooth, Piecewise Smooth and Smoothing Methods (M. Fukushima, L. Qi, eds.). Kluwer Academic Publishers, Boston, 1999, pp. 45–63.

[2] *J. Burke, S. Xu*: A non-interior predictor-corrector path following algorithm for the monotone linear complementarity problem. Math. Program. *87* (2000), 113–130.

[3] *B. Chen, X. Chen*: A global and local superlinear continuation-smoothing method for $P_0 + R_0$ and monotone NCP. SIAM J. Optim. *9* (1999), 624–645.

[4] *B. Chen, X. Chen*: A global linear and local quadratic continuation smoothing method for variational inequalities with box constraints. Comput. Optim. Appl. *17* (2000), 131–158.

[5] *B. Chen, N. Xiu*: A global linear and local quadratic noninterior continuation method for nonlinear complementarity problems based on Chen-Mangasarian smoothing functions. SIAM J. Optim. *9* (1999), 605–623.

[6] *J. Chen*: A new merit function and its related properties for the second-order cone complementarity problem. Pac. J. Optim. *2* (2006), 167–179.

[7] *J. Chen, X. Chen, P. Tseng*: Analysis of nonsmooth vector-valued functions associated with second-order cones. Math. Program. *101* (2004), 95–117.

[8] *J. Chen, P. Tseng*: An unconstrained smooth minimization reformulation of the second-order cone complementarity problem. Math. Program. *104* (2005), 293–327.

[9] *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.

[10] *X. N. Chi, S. Y. Liu*: Analysis of a non-interior continuation method for second-order cone programming. J. Appl. Math. Comput. *27* (2008), 47–61.

[11] *X. N. Chi, S. Y. Liu*: A one-step smoothing Newton method for second-order cone programming. J. Comput. Appl. Math. *223* (2009), 114–123.

[12] *X. N. Chi, S. Y. Liu*: A non-interior continuation method for second-order cone programming. Optimization *58* (2009), 965–979.

[13] *F. H. Clarke*: Optimization and Nonsmooth Analysis. John Wiley & Sons, New York, 1983, reprinted by SIAM, Philadelphia, 1990.

[14] *L. Fang*: A new one-step smoothing Newton method for nonlinear complementarity problem with P_0-function. Appl. Math. Comput. *216* (2010), 1087–1095.

[15] *L. Fang, G. P. He, Y. H. Hu*: A new smoothing Newton-type method for second-order cone programming problems. Appl. Math. Comput. *215* (2009), 1020–1029.

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

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

[18] *Z. H. Huang, J. Y. Han, D. C. Xu, L. P. Zhang*: The non-interior continuation methods for solving the P_0 function nonlinear complementarity problem. Sci. China, Ser. A *44* (2001), 1107–1114.

[19] *Z. H. Huang, T. Ni*: Smoothing algorithms for complementarity problems over symmetric cones. Comput. Optim. Appl. *45* (2010), 557–579.

[20] *C. Ma, X. Chen*: The convergence of a one-step smoothing Newton method for P_0-NCP based on a new smoothing NCP-function. J. Comput. Appl. Math. *216* (2008), 1–13.

[21] *R. Mifflin*: Semismooth and semiconvex functions in constrained optimization. SIAM J. Control. Optim. *15* (1977), 957–972.

[22] *S. H. Pan, J. S. Chen*: A damped Gauss-Newton method for the second-order cone complementarity problem. Appl. Math. Optim. *59* (2009), 293–318.

[23] *S. H. Pan, J. S. Chen*: A linearly convergent derivative-free descent method for the second-order cone complementarity problem. Optimization *59* (2010), 1173–1197.

[24] *L. Qi*: Convergence analysis of some algorithms for solving nonsmooth equations. Math. Oper. Res. *18* (1993), 227–244.

[25] *L. Qi, J. Sun*: A nonsmooth version of Newton's method. Math. Program. *58* (1993), 353–367.

[26] *L. Qi, D. Sun*: Improving the convergence of non-interior point algorithm for nonlinear complementarity problems. Math. Comput. *69* (2000), 283–304.

[27] *L. Qi, D. Sun, G. Zhou*: A new look at smoothing Newton methods for nonlinear complementarity problems and box constrained variational inequalities. Math. Program. *87* (2000), 1–35.

[28] *J. Y. Tang, G. P. He, L. Dong, L. Fang*: A smoothing Newton method for second-order cone optimization based on a new smoothing function. Appl. Math. Comput. *218* (2011), 1317–1329.

[29] *K. C. Toh, R. H. Tütüncü, M. J. Todd*: SDPT3 Version 3.02-A MATLAB software for semidefinite-quadratic-linear programming, 2000. `http://www.math.nus.edu.sg/~mattohkc/sdpt3.html`.

[30] *A. Yoshise*: Interior point trajectories and a homogeneous model for nonlinear complementarity problems over symmetric cones. SIAM J. Optim. *17* (2006), 1129–1153.

[31] *L. Zhang, J. Han, Z. Huang*: Superlinear/quadratic one-step smoothing Newton method for P_0-NCP. Acta Math. Sin. *21* (2005), 117–128.

[32] *J. Zhang, K. Zhang*: A variant smoothing Newton method for P_0-NCP based on a new smoothing function. J. Comput. Appl. Math. *225* (2009), 1–8.

[33] *G. Zhou, D. Sun, L. Qi*: Numerical experiments for a class of squared smoothing Newton methods for box constrained variational inequality problems. Reformulation-Nonsmooth, Piecewise Smooth, Semismooth and Smoothing Methods (M. Fukushima, L. Qi, eds.). Kluwer Academic Publishers, Boston, 1999, pp. 421–441.

Authors' addresses: *J. Tang* (corresponding author), College of Mathematics and Information Science, Xinyang Normal University, 464000 Xinyang, P. R. China, and Department of Mathematics, Shanghai Jiaotong University, 200240 Shanghai, P. R. China, e-mail: `tangjingyong926@163.com`; *G. He*, College of Information Science and Engineering, Shandong University of Science and Technology, 266510 Qingdao, P. R. China, e-mail: `hegp@263.net`; *L. Dong*, College of Mathematics and Information Science, Xinyang Normal University, 464000 Xinyang, P. R. China, e-mail: `citycity926@163.com`; *L. Fang*, College of Mathematics and Systems Science, Taishan University, 271012 Tai'an, P. R. China, e-mail: `fangliang3@163.com`; *J. Zhou*, Department of Mathematics, School of Science, Shandong University of Technology, 255049 Zibo, P. R. China, e-mail: `jinchuanzhou@163.com`.