Applied Numerical Mathematics 120 (2017) 82–96

Contents lists available at ScienceDirect

Applied Numerical Mathematics

www.elsevier.com/locate/apnum

A smoothing Newton method for absolute value equation associated with second-order cone

Xin-He Miao [a,1], Jian-Tao Yang [a], B. Saheya [b,2], Jein-Shan Chen [c,*,3]

[a] *Department of Mathematics, Tianjin University, China, Tianjin 300072, China*
[b] *College of Mathematical Science, Inner Mongolia Normal University, Hohhot 010022, Inner Mongolia, PR China*
[c] *Department of Mathematics, National Taiwan Normal University, Taipei 11677, Taiwan*

A R T I C L E   I N F O

Article history:
Received 20 May 2016
Received in revised form 26 February 2017
Accepted 28 April 2017
Available online 4 May 2017

Keywords:
Second-order cone
Absolute value equations
Smoothing Newton algorithm

A B S T R A C T

In this paper, we consider the smoothing Newton method for solving a type of absolute value equations associated with second order cone (SOCAVE for short), which is a generalization of the standard absolute value equation frequently discussed in the literature during the past decade. Based on a class of smoothing functions, we reformulate the SOCAVE as a family of parameterized smooth equations, and propose the smoothing Newton algorithm to solve the problem iteratively. Moreover, the algorithm is proved to be locally quadratically convergent under suitable conditions. Preliminary numerical results demonstrate that the algorithm is effective. In addition, two kinds of numerical comparisons are presented which provides numerical evidence about why the smoothing Newton method is employed and also suggests a suitable smoothing function for future numerical implementations. Finally, we point out that although the main idea for proving the convergence is similar to the one used in the literature, the analysis is indeed more subtle and involves more techniques due to the feature of second-order cone.

1. Introduction

The standard absolute value equation (AVE) is in the form of

$$Ax + B|x| = b, \tag{1}$$

where $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times n}$, $B \neq 0$, and $b \in \mathbb{R}^n$. Here $|x|$ means the componentwise absolute value of vector $x \in \mathbb{R}^n$. When $B = -I$, where I is the identity matrix, the AVE (1) reduces to the special form:

$$Ax - |x| = b.$$

It is known that the AVE (1) was first introduced by Rohn in [38] and recently has been investigated by many researchers, for example, Caccetta, Qu and Zhou [1], Hu and Huang [14], Jiang and Zhang [22], Ketabchi and Moosaei [23], Mangasarian [25–32], Mangasarian and Meyer [34], Prokopyev [35], and Rohn [40].

* Corresponding author.
E-mail addresses: xinhemiao@tju.edu.cn (X.-H. Miao), zzlyyjt@163.com (J.-T. Yang), saheya@imnu.edu.cn (B. Saheya), jschen@math.ntnu.edu.tw (J.-S. Chen).

1 The author's work is supported by National Natural Science Foundation of China (No. 11471241).
2 The author's work is supported by Natural Science Foundation of Inner Mongolia (Award Number: 2014MS0119).
3 The author's work is supported by Ministry of Science and Technology, Taiwan (No. 104-2115-M-003-011-MY2).

http://dx.doi.org/10.1016/j.apnum.2017.04.012
0168-9274/

In particular, Mangasarian and Meyer [34] show that the AVE (1) is equivalent to the bilinear program, the generalized LCP (linear complementarity problem), and the standard LCP provided 1 is not an eigenvalue of A. With these equivalent reformulations, they also show that the AVE (1) is NP-hard in its general form and provide existence results. Prokopyev [35] further improves the above equivalence which indicates that the AVE (1) can be equivalently recast as LCP without any assumption on A and B, and also provides a relationship with mixed integer programming. In general, if solvable, the AVE (1) can have either unique solution or multiple (e.g., exponentially many) solutions. Indeed, various sufficiency conditions on solvability and non-solvability of the AVE (1) with unique and multiple solutions are discussed in [34,35,39]. Some variants of the AVE, like the absolute value equation associated with second-order cone and the absolute value programs, are investigated in [16] and [41], respectively.

In this paper, we target another type of absolute value equation which is a natural extension of the standard AVE (1). More specifically the following absolute value equation associated with second-order cones, abbreviated as SOCAVE, as below:

$$Ax + B|x| = b, \tag{2}$$

where $A, B \in \mathbb{R}^{n\times n}$ and $b \in \mathbb{R}^n$ are the same as those in (1); $|x|$ denotes the absolute value of x coming from the square root of the Jordan product "$\circ$" of x and x. What is the difference between the standard AVE (1) and the SOCAVE (2)? Their mathematical formats look the same. In fact, the main difference is that $|x|$ in the standard AVE (1) means the componentwise $|x_i|$ of each $x_i \in \mathbb{R}$, i.e., $|x| = (|x_1|, |x_2|, \cdots, |x_n|)^T \in \mathbb{R}^n$; however, $|x|$ in the SOCAVE (2) denotes the vector satisfying $\sqrt{x^2} := \sqrt{x \circ x}$ associated with second-order cone under Jordan product. To understand its meaning, we need to introduce the definition of second-order cone (SOC). The second-order cone in $\mathbb{R}^n$ ($n \ge 1$), also called the Lorentz cone, is defined as

$$\mathcal{K}^n := \left\{ (x_1, x_2) \in \mathbb{R} \times \mathbb{R}^{n-1} \mid \|x_2\| \le x_1 \right\},$$

where $\|\cdot\|$ denotes the Euclidean norm. If $n = 1$, then $\mathcal{K}^n$ is the set of nonnegative reals $\mathbb{R}_+$. In general, a general second-order cone $\mathcal{K}$ could be the Cartesian product of SOCs, i.e.,

$$\mathcal{K} := \mathcal{K}^{n_1} \times \cdots \times \mathcal{K}^{n_r}.$$

For simplicity, we focus on the single SOC $\mathcal{K}^n$ because all the analysis can be carried over to the setting of Cartesian product. The SOC is a special case of symmetric cones and can be analyzed under Jordan product, see [11]. In particular, for any two vectors $x = (x_1, x_2) \in \mathbb{R} \times \mathbb{R}^{n-1}$ and $y = (y_1, y_2) \in \mathbb{R} \times \mathbb{R}^{n-1}$, the *Jordan product* of x and y associated with $\mathcal{K}^n$ is defined as

$$x \circ y := \begin{bmatrix} x^T y \\ y_1 x_2 + x_1 y_2 \end{bmatrix}.$$

The Jordan product, unlike scalar or matrix multiplication, is not associative, which is a main source of complication in the analysis of optimization problems involved SOC, see [3,10,12] and references therein for more details. The identity element under this Jordan product is $e = (1, 0, \ldots, 0)^T \in \mathbb{R}^n$. With these definitions, x^2 means the Jordan product of x with itself, i.e., $x^2 := x \circ x$; and $\sqrt{x}$ with $x \in \mathcal{K}^n$ denotes the unique vector such that $\sqrt{x} \circ \sqrt{x} = x$. In other words, the vector $|x|$ in the SOCAVE (2) is computed by

$$|x| := \sqrt{x \circ x}.$$

As mentioned earlier, the significance of the AVE (1) arises from the fact that the AVE is capable to formulate many optimization problems (also see [26,30,32,34,35]), such as, linear programs, quadratic programs, bimatrix games, and so on. Moreover, the absolute value equations is equivalent to the linear complementarity problem [34]. Accordingly, we see that the SOCAVE (2) plays similar role in various optimization problems involved second-order cones. For solving the standard AVE (1), there are many various numerical methods proposed in the literature (see [1,21,22,25–27,35,43]). As for the SOCAVE (2), Hu, Huang and Zhang [16] propose a generalized Newton method for solving the SOCAVE (2). It is well known that smoothing-type algorithms is a powerful tool for solving many optimization problems, for example, the linear and nonlinear complementarity problems [3,12,19,20,24], the system of equalities and inequalities [17,42]. In this paper, we are interested in a smoothing Newton method for solving the SOCAVE (2). Our numerical results also support that the smoothing Newton method is a better way than the generalized Newton method employed in [16]. That is why we adopt this algorithm as the main tool to do numerical implementations. In addition, we have shown that the proposed smoothing Newton method is locally quadratically convergent under suitable condition. We report some preliminary numerical results to show that the method is efficient. Moreover, numerical comparisons based on various value of p are presented as well.

To close this section, we say a few words about notations and the organization of this paper. As usual, $\mathbb{R}^n$ denotes the space of n-dimensional real column vectors. $\mathbb{R}_+$ and $\mathbb{R}_{++}$ denote the nonnegative and positive reals. For any $x, y \in \mathbb{R}^n$, the Euclidean inner product is denoted $\langle x, y\rangle = x^T y$, and the Euclidean norm $\|x\|$ is denoted as $\|x\| = \sqrt{\langle x, x\rangle}$. This paper is organized as follows. In Section 2, we briefly describe some concepts and properties on second-order cone. Besides, we review Jordan product and the spectral decomposition for elements x and y in $\mathbb{R}^n$. In Section 3, we introduce a smoothing function of the absolute value $|x|$, and study the Jacobian matrix of the smoothing function. In Section 4, we propose a smoothing Newton algorithm for solving the SOCAVE (2), and discuss the convergence of the proposed method under suitable conditions. In Section 5, the preliminary numerical results and numerical comparisons are given.

2. Preliminaries

In this section, we recall some basic concepts and background materials regarding the second-order cone, which will be extensively used in the subsequent analysis. More details can be found in [3,10–12,16]. First, we recall the expression of the *spectral decomposition* of x with respect to SOC. For $x = (x_1, x_2) \in \mathbb{R} \times \mathbb{R}^{n-1}$, the spectral decomposition of x with respect to SOC is given by

$$x = \lambda_1(x) u_x^{(1)} + \lambda_2(x) u_x^{(2)}, \tag{3}$$

where $\lambda_i(x) = x_1 + (-1)^i \|x_2\|$ for $i = 1, 2$ and

$$u_x^{(i)} = \begin{cases} \frac{1}{2}\left(1, (-1)^i \frac{x_2^T}{\|x_2\|}\right)^T & \text{if} \quad \|x_2\| \neq 0, \\ \frac{1}{2}\left(1, (-1)^i \omega^T\right)^T & \text{if} \quad \|x_2\| = 0, \end{cases} \tag{4}$$

with $\omega \in \mathbb{R}^{n-1}$ being any vector satisfying $\|\omega\| = 1$. The two scalars $\lambda_1(x)$ and $\lambda_2(x)$ are called spectral values of x; while the two vectors $u_x^{(1)}$ and $u_x^{(2)}$ are called the spectral vectors of x. Moreover, it is obvious that the spectral decomposition of $x \in \mathbb{R}^n$ is unique if $x_2 \neq 0$.

Lemma 2.1. *For any $x = (x_1, x_2) \in \mathbb{R} \times \mathbb{R}^{n-1}$ with the spectral decomposition given as in (3)-(4), the following results hold.*

(a) $u_x^{(1)} \circ u_x^{(2)} = 0$ *and* $u_x^{(i)} \circ u_x^{(i)} = u_x^{(i)}$ *for* $i = 1, 2$;
(b) $\|u_x^{(1)}\|^2 = \|u_x^{(2)}\|^2 = \frac{1}{2}$ *and* $\|x\|^2 = \frac{1}{2}(\lambda_1^2(x) + \lambda_2^2(x))$.

Proof. The property can be verified directly or can be found in [3,11,12,16,10]. □

In the next content, we talk about the projection onto second-order cone. We let x_+ be the projection of x onto SOC $\mathcal{K}^n$, and x_- be the projection of $-x$ onto the dual cone $(\mathcal{K}^n)^*$ of $\mathcal{K}^n$, where the dual cone $(\mathcal{K}^n)^*$ is defined by $(\mathcal{K}^n)^* := \{y \in \mathbb{R}^n \mid \langle x, y \rangle \geq 0, \forall x \in \mathcal{K}^n\}$. In fact, the dual cone of $\mathcal{K}^n$ is itself, i.e., $(\mathcal{K}^n)^* = \mathcal{K}^n$. Due to the special structure of SOC $\mathcal{K}^n$, the explicit formula of projection of $x = (x_1, x_2) \in \mathbb{R} \times \mathbb{R}^{n-1}$ onto $\mathcal{K}^n$ is obtained in [3,10–13] as below:

$$x_+ = \begin{cases} x & \text{if } x \in \mathcal{K}^n, \\ 0 & \text{if } x \in -\mathcal{K}^n, \\ u & \text{otherwise,} \end{cases}$$

where

$$u = \begin{bmatrix} \frac{x_1 + \|x_2\|}{2} \\ \left(\frac{x_1 + \|x_2\|}{2}\right) \frac{x_2}{\|x_2\|} \end{bmatrix}.$$

Similarly, the expression of x_- is in the form of

$$x_- = \begin{cases} 0 & \text{if } x \in \mathcal{K}^n, \\ -x & \text{if } x \in -\mathcal{K}^n, \\ w & \text{otherwise,} \end{cases}$$

where

$$w = \begin{bmatrix} -\frac{x_1 - \|x_2\|}{2} \\ \left(\frac{x_1 - \|x_2\|}{2}\right) \frac{x_2}{\|x_2\|} \end{bmatrix}.$$

Together with the spectral decomposition of x, it is shown that $x = x_+ + x_-$ and the expression of x_+ has the form:

$$x_+ = (\lambda_1(x))_+ u_x^{(1)} + (\lambda_2(x))_+ u_x^{(2)},$$

and

$$x_- = (-\lambda_1(x))_+ u_x^{(1)} + (-\lambda_2(x))_+ u_x^{(2)},$$

where $(\alpha)_+ = \max\{0, \alpha\}$ for $\alpha \in \mathbb{R}$.

Next, we talk about the expression of $|x|$ associated with SOC. There is an alternative way via the so-called SOC-function to obtain the expression of $|x|$, which can be found in [2,4]. More specifically, for any $x \in \mathbb{R}^n$, we define the **absolute value** $|x|$ of x with respect to SOC as $|x| := x_+ + x_-$. In fact, in the setting of SOC, the form $|x| = x_+ + x_-$ is equivalent to the form

$|x| = \sqrt{x \circ x}$. Combining the above expression of x_+ and x_-, it cab be verified that the expression of the absolute value $|x|$ is in the form of

$$\begin{aligned} |x| &= \left[(\lambda_1(x))_+ + (-\lambda_1(x))_+\right] u_x^{(1)} + \left[(\lambda_2(x))_+ + (-\lambda_2(x))_+\right] u_x^{(2)} \\ &= \left|\lambda_1(x)\right| u_x^{(1)} + \left|\lambda_2(x)\right| u_x^{(2)}. \end{aligned}$$

To end this section, we point out the relation between SOCAVE and SOCLCP (second-order cone linear complementarity problem). In [16], it was shown that SOCAVE (2) is equivalent to the following SOCLCP: find $x, y \in \mathbb{R}^n$ such that

$$Mx + Py = c, \quad \text{and} \quad x \in \mathcal{K}^n,\ y \in \mathcal{K}^n,\ \langle x, y \rangle = 0,$$

where $M, P \in \mathbb{R}^{n\times n}$ are matrices and $c \in \mathbb{R}^n$. However, the above is not a standard SOCLCP because there exists the equations $Mx + Py = c$ therein. As below, we show that the SOCAVE (2) can be further converted into a standard SOCLCP.

Theorem 2.1. *The SOCAVE (2) can be reduced to the second-order cone linear complementarity problem (SOCLCP):*

$$v \in \mathcal{K}^n \times \mathcal{K}^n \times \mathcal{K}^n,\ w = Qv + q \in \mathcal{K}^n \times \mathcal{K}^n \times \mathcal{K}^n \quad \text{and} \quad \langle v, w \rangle = 0, \tag{5}$$

where

$$Q := \begin{bmatrix} -I & 2I & 0 \\ A & B-A & 0 \\ -A & A-B & 0 \end{bmatrix}, \quad v := \begin{bmatrix} 2x_+ \\ |x| \\ 0 \end{bmatrix} \ \text{and}\ q := \begin{bmatrix} 0 \\ -b \\ b \end{bmatrix}. \tag{6}$$

Proof. By looking into (6), we have

$$w = Qv + q = \begin{bmatrix} 2x_- \\ Ax + B|x| - b \\ -Ax - B|x| + b \end{bmatrix}.$$

Plugging this into SOCLCP (5) implies that

$$Ax + B|x| - b \in \mathcal{K}^n \quad \text{and} \quad -Ax - B|x| + b \in \mathcal{K}^n.$$

Since $\mathcal{K}^n$ is pointed, it follows that $Ax + B|x| - b = 0$. On the other hand, the above argument is reversible. Thus, we show that SOCAVE (2) is equivalent to second-order cone linear complementarity problem. □

Remark 2.1. From Theorem 2.1, it follows that we can also solve the SOCAVE (2) by employing many efficient algorithms for solving SOCLCP (5). Nonetheless, when we apply the Newton method to solve SOCLCP, it still needs reformulate it as smooth equations or nonsmooth equations. This means that we need twice reformulations if we follow this way. In view of this, in this paper, we reformulate the SOCAVE (2) directly as the smooth equations, and solve the equations by smoothing Newton method.

3. Smoothing functions associate with SOCAVE

In this paper, we employ the smoothing Newton method for solving the SOCAVE (2). To this end, we need to adopt a smoothing function. Due to the non-differentiability of $|\alpha|$ for $\alpha \in \mathbb{R}$, we consider a class of smoothing functions for the absolute value function $|\alpha|$. More specifically, we define the function $\phi_p(\cdot,\cdot): \mathbb{R}^2 \to \mathbb{R}$ as

$$\phi_p(a, b) := \sqrt[p]{|a|^p + |b|^p}, \quad p > 1. \tag{7}$$

This class of functions is extracted from the so-called generalized Fischer–Burmeister function $\phi_p(a, b) = \sqrt[p]{|a|^p + |b|^p} - (a + b)$, which is heavily studied in many references [5–9,15]. For convenience, we still use the notation ϕ_p even it is no longer exactly the same as the generalized Fischer–Burmeister function.

Lemma 3.1. *Let $\phi_p: \mathbb{R}^2 \to \mathbb{R}$ be defined as in (7). Then, the following hold.*

(a) *$\phi_p(a, 0) = |a|$ and $\phi_p(0, b) = |b|$;*
(b) *$\phi_p(\cdot,\cdot)$ is Lipschitz continuous on $\mathbb{R}^2$;*
(c) *$\phi_p(\cdot,\cdot)$ is strongly semismooth on $\mathbb{R}^2$;*

(d) *$\phi_p(a,b)$ is continuously differentiable for any $(a,b) \neq (0,0) \in \mathbb{R}^2$ with*

$$\frac{\partial \phi_p(a,b)}{\partial a} = \frac{\operatorname{sgn}(a)|a|^{p-1}}{(\phi_p(a,b))^{p-1}} \quad \text{and} \quad \frac{\partial \phi_p(a,b)}{\partial b} = \frac{\operatorname{sgn}(b)|b|^{p-1}}{(\phi_p(a,b))^{p-1}},$$

where the function $\operatorname{sgn}(\cdot)$ *is defined by* $\operatorname{sgn}(\alpha) := \begin{cases} 1 & \text{if } \alpha > 0, \\ 0 & \text{if } \alpha = 0, \\ -1 & \text{if } \alpha < 0. \end{cases}$

Proof. Please refer to [5–9,15] for a proof. □

According to Lemma 3.1, it follows that for any $a \in \mathbb{R}$ and $a \to 0$, we have $\phi_p(a,b) \to |b|$. Therefore, combining the spectral decomposition of x and the function ϕ_p, we define a vector-valued smoothing function $\Phi_p : \mathbb{R} \times \mathbb{R}^n \to \mathbb{R}^n$ as

$$\begin{aligned} \Phi_p(\mu, x) &= \phi_p(\mu, \lambda_1(x)) u_x^{(1)} + \phi_p(\mu, \lambda_2(x)) u_x^{(2)} \\ &= \sqrt[p]{|\mu|^p + |\lambda_1(x)|^p}\, u_x^{(1)} + \sqrt[p]{|\mu|^p + |\lambda_2(x)|^p}\, u_x^{(2)}, \end{aligned}$$

where $\mu \in \mathbb{R}$ is a parameter, and $\lambda_1(x), \lambda_2(x)$ are the spectral values of x. From Lemma 3.1, it is easy to verify that

$$\lim_{\mu \to 0} \Phi_p(\mu, x) = |\lambda_1(x)|\, u_x^{(1)} + |\lambda_2(x)|\, u_x^{(2)} = |x|.$$

In other words, the function $\Phi_p(\mu, x)$ is a uniformly smoothing function of $|x|$ associated with SOC. With this function, for the SOCAVE (2), we further define a function $H(\mu, x) : \mathbb{R} \times \mathbb{R}^n \to \mathbb{R} \times \mathbb{R}^n$ by

$$H(\mu, x) = \begin{bmatrix} \mu \\ Ax + B\Phi_p(\mu, x) - b \end{bmatrix}, \quad \forall \mu \in \mathbb{R},\ x \in \mathbb{R}^n. \tag{8}$$

Then, we observe that

$$\begin{aligned} H(\mu, x) = 0 &\Longleftrightarrow \mu = 0 \ \text{ and } \ Ax + B\Phi_p(\mu, x) - b = 0 \\ &\Longleftrightarrow Ax + B|x| - b = 0 \ \text{ and } \ \mu = 0. \end{aligned}$$

This indicates that x is a solution to the SOCAVE (2) if and only if (μ, x) is a solution to the equation $H(\mu, x) = 0$. In fact, we often choose $\mu \in \mathbb{R}_{++}$. Applying Lemma 3.1 again, it is not difficult to show that the function $H(\mu, x)$ is continuously differentiable on $\mathbb{R}_{++} \times \mathbb{R}^n$. From direct calculation, we can also obtain the explicit formula of the Jacobian matrix for the function H as below:

$$H'(\mu, x) = \begin{bmatrix} 1 & 0 \\ B\frac{\partial \Phi_p(\mu,x)}{\partial \mu} & A + B\frac{\partial \Phi_p(\mu,x)}{\partial x} \end{bmatrix} \tag{9}$$

for all $(\mu, x) \in \mathbb{R}_{++} \times \mathbb{R}^n$ with $x = (x_1, x_2) \in \mathbb{R} \times \mathbb{R}^{n-1}$, where

$$\begin{aligned} \frac{\partial \Phi_p(\mu, x)}{\partial \mu} &= \frac{\partial \phi_p(\mu, \lambda_1(x))}{\partial \mu} u_x^{(1)} + \frac{\partial \phi_p(\mu, \lambda_2(x))}{\partial \mu} u_x^{(2)} \\ &= \frac{\mu^{p-1}}{[\phi_p(\mu, \lambda_1(x))]^{p-1}} u_x^{(1)} + \frac{\mu^{p-1}}{[\phi_p(\mu, \lambda_2(x))]^{p-1}} u_x^{(2)} \end{aligned}$$

and

$$\frac{\partial \Phi_p(\mu, x)}{\partial x} = \begin{cases} \frac{\operatorname{sgn}(x_1)|x_1|^{p-1}}{\left[\sqrt[p]{\mu^p + |x_1|^p}\right]^{p-1}} I & \text{if } x_2 = 0, \\ \begin{bmatrix} b & c\frac{x_2^T}{\|x_2\|} \\ c\frac{x_2}{\|x_2\|} & aI + (b-a)\frac{x_2 x_2^T}{\|x_2\|^2} \end{bmatrix} & \text{if } x_2 \neq 0, \end{cases}$$

with

$$\begin{aligned} a &= \frac{\phi_p(\mu, \lambda_2(x)) - \phi_p(\mu, \lambda_1(x))}{\lambda_2(x) - \lambda_1(x)}, \\ b &= \frac{1}{2}\left(\frac{\operatorname{sgn}(\lambda_2(x))|\lambda_2(x)|^{p-1}}{[\phi_p(\mu, \lambda_2(x))]^{p-1}} + \frac{\operatorname{sgn}(\lambda_1(x))|\lambda_1(x)|^{p-1}}{[\phi_p(\mu, \lambda_1(x))]^{p-1}} \right), \\ c &= \frac{1}{2}\left(\frac{\operatorname{sgn}(\lambda_2(x))|\lambda_2(x)|^{p-1}}{[\phi_p(\mu, \lambda_2(x))]^{p-1}} - \frac{\operatorname{sgn}(\lambda_1(x))|\lambda_1(x)|^{p-1}}{[\phi_p(\mu, \lambda_1(x))]^{p-1}} \right). \end{aligned} \tag{10}$$

4. Smoothing Newton method

In this section, we investigate the smoothing algorithm based on the smoothing function $\Phi_p(\mu, x)$ for solving the SOCAVE (2), and show the convergence properties of the considered algorithm. First, we present the generic framework of the smoothing algorithm.

Algorithm 4.1 *(A smoothing Newton algorithm).*

Step 0 Choose $\delta \in (0,1)$, $\sigma \in (0,1)$, and $\mu_0 \in \mathbb{R}_{++}$, $x^0 \in \mathbb{R}^n$. Set $z^0 := (\mu_0, x^0)$, $e := (1,0) \in \mathbb{R} \times \mathbb{R}^{n-1}$. Choose $\beta > 1$ satisfying $\min\{1, \|H(z^0)\|^2\} \le \beta\mu_0$. Set $k := 0$.

Step 1 If $\|H(z^k)\| = 0$, stop. Otherwise, set $\tau_k := \min\{1, \|H(z^k)\|\}$.

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

$$H(z^k) + H'(z^k)\triangle z^k = \frac{1}{\beta}\tau_k^2 e, \tag{11}$$

where $H'(z^k)$ denotes the Jacobian matrix of $H(z^k)$ at (μ_k, x^k) given by (9).

Step 3 Let α_k be the maximum of the values $1, \delta, \delta^2, \cdots$ such that

$$\|H(z^k + \alpha_k\triangle z^k)\| \le \left[1 - \sigma(1 - \frac{1}{\beta})\alpha_k\right]\|H(z^k)\|. \tag{12}$$

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

In order to explain that Algorithm 4.1 is well defined, we have to prove that the system of Newton equation (11) is solvable, and the line search (12) is well-defined. To this end, we need the next two technical lemmas.

Lemma 4.1. *For any $M, N \in \mathbb{R}^{n\times n}$, $\sigma_{\min}(M) > \sigma_{\max}(N)$ if and only if $\sigma_{\min}(M^TM) > \sigma_{\max}(N^TN)$. In addition, if $\sigma_{\min}(M^TM) > \sigma_{\max}(N^TN)$, then $M^TM - N^TN$ is positive definite. Here $\sigma_{\min}(M)$ denotes the minimum singular value of M, and $\sigma_{\max}(N)$ denotes the maximum singular value of N.*

Proof. The proof is straightforward or can be found in usual textbook of matrix analysis, so we omit it here. □

Lemma 4.2. *Let $A, S \in \mathbb{R}^{n\times n}$ and A be symmetric. Suppose that the eigenvalues of A and SS^T are arranged in non-increasing order. Then, for each $k = 1, 2, \cdots, n$, there exists a nonnegative real number θ_k such that*

$$\lambda_{\min}(SS^T) \le \theta_k \le \lambda_{\max}(SS^T) \quad \text{and} \quad \lambda_k(SAS^T) = \theta_k\lambda_k(A).$$

Proof. Please see [18, Corollary 4.5.11] for a proof. □

In order to show that the Jacobian matrix $H'(\mu, x)$ in Newton equation (11) is nonsingular for any $\mu > 0$. We need the following assumption:

Assumption 4.1. For the SOCAVE (2), it holds $\sigma_{\min}(A) > \sigma_{\max}(B)$.

In fact, under the condition of Assumption 4.1, the SOCAVE (2) has a unique solution, which is verified in [33].

Theorem 4.1. *Let H be defined as in (8). Suppose that Assumption 4.1 holds. Then, the Jacobian matrix $H'(\mu, x)$ in Newton equations (11) is nonsingular for any $\mu > 0$.*

Proof. From the expression of $H'(\mu, x)$ given as in (9), we know that $H'(\mu, x)$ is nonsingular if and only if the matrix $A + B\frac{\partial\Phi(\mu,x)}{\partial x}$ is nonsingular. Thus, it suffices to show that the matrix $A + B\frac{\partial\Phi(\mu,x)}{\partial x}$ is nonsingular. Suppose not, i.e., there exists a vector $0 \ne v \in \mathbb{R}^n$ such that

$$\left[A + B\frac{\partial\Phi(\mu,x)}{\partial x}\right]v = 0.$$

This implies that

$$v^TA^TAv = v^T\left[\frac{\partial\Phi(\mu,x)}{\partial x}\right]^T B^TB\,\frac{\partial\Phi(\mu,x)}{\partial x}\,v. \tag{13}$$

For convenience, we denote $C := \frac{\partial \Phi(\mu,x)}{\partial x}$. Then, it follows that $v^T A^T A v = v^T C^T B^T B C v$. By Lemma 4.2, there exists a constant $\hat{\theta}$ such that

$$\lambda_{\min}(C^T C) \le \hat{\theta} \le \lambda_{\max}(C^T C) \quad \text{and} \quad \lambda_{\max}(C^T B^T B C) = \hat{\theta}\lambda_{\max}(B^T B).$$

Note that if we can prove that $0 \le \lambda_{\min}(C^T C) \le \lambda_{\max}(C^T C) \le 1$, we have $\lambda_{\max}(C^T B^T B C) \le \lambda_{\max}(B^T B)$. Then, by the assumption that the minimum singular value of A strictly exceeds the maximum singular value of B, and applying Lemma 4.1, we obtain $v^T A^T A v > v^T C^T B^T B C v$. This contradicts the formula (13), which shows the Jacobian matrix $H'(\mu, x)$ in Newton equations (11) is nonsingular for $\mu > 0$.

Thus, as discussed above, we only need to prove $0 \le \lambda_{\min}(C^T C) \le \lambda_{\max}(C^T C) \le 1$. For $x_2 = 0$, we compute that $C = \frac{\mathrm{sgn}(x_1)|x_1|^{p-1}}{[\sqrt[p]{\mu^p + |x_1|^p}]^{p-1}} I$. Then, it is clear that $0 < \lambda(C^T C) < 1$ for $\mu > 0$. For $x_2 \neq 0$, using the fact that the matrix $M^T M$ is always positive semidefinite for any matrix $M \in \mathbb{R}^{m\times n}$, we see that the inequality $\lambda_{\min}(C^T C) \ge 0$ always holds. In order to prove that $\lambda_{\max}(C^T C) \le 1$, we need to further prove that the matrix $I - C^T C$ is positive semidefinite. To see this, note that

$$I - C^T C = \begin{bmatrix} 1 - b^2 - c^2 & -2bc\frac{x_2^T}{\|x_2\|} \\ -2bc\frac{x_2}{\|x_2\|} & (1-a^2)I + (a^2 - b^2 - c^2)\frac{x_2 x_2^T}{\|x_2\|^2} \end{bmatrix}.$$

Because $b^2 + c^2 = \frac{1}{2}\left[\frac{|\lambda_2(x)|^{2(p-1)}}{[\phi_p(\mu,\lambda_2(x))]^{2(p-1)}} + \frac{|\lambda_1(x)|^{2(p-1)}}{[\phi_p(\mu,\lambda_1(x))]^{2(p-1)}}\right] < \frac{1}{2}\cdot 2 = 1$ for $\mu > 0$, we have $1 - b^2 - c^2 > 0$. Moreover, the Schur complement of $1 - b^2 - c^2$ has the form of

$$\begin{aligned} &(1-a^2)I + (a^2 - b^2 - c^2)\frac{x_2 x_2^T}{\|x_2\|^2} - \frac{4b^2c^2}{1-b^2-c^2}\frac{x_2 x_2^T}{\|x_2\|^2} \\ &= (1-a^2)\left(I - \frac{x_2 x_2^T}{\|x_2\|^2}\right) + \left(1 - b^2 - c^2 - \frac{4b^2c^2}{1-b^2-c^2}\right)\frac{x_2 x_2^T}{\|x_2\|^2}. \end{aligned} \tag{14}$$

On the other hand, $|\lambda_i(x)| < \phi_p(\mu, \lambda_i(x)) \quad (i = 1, 2)$ for $\mu > 0$, we have

$$\begin{aligned} &\left|\phi_p(\mu,\lambda_2(x)) - \phi_p(\mu,\lambda_1(x))\right| \\ &= \left|\frac{|\lambda_2(x)|^p - |\lambda_1(x)|^p}{\sum_{i=1}^{p}\left[\phi_p(\mu,\lambda_2(x))\right]^{p-i}\left[\phi_p(\mu,\lambda_1(x))\right]^{i-1}}\right| \\ &= \left|\frac{(|\lambda_2(x)| - |\lambda_1(x)|)\sum_{i=1}^{p}|\lambda_2(x)|^{p-i}|\lambda_1(x)|^{i-1}}{\sum_{i=1}^{p}\left[\phi_p(\mu,\lambda_2(x))\right]^{p-i}\left[\phi_p(\mu,\lambda_1(x))\right]^{i-1}}\right| \\ &< ||\lambda_2(x)| - |\lambda_1(x)|| \\ &\le |\lambda_2(x) - \lambda_1(x)|. \end{aligned}$$

This together with (10) implies that $1 - a^2 > 0$ for any $\mu > 0$. In addition, for any $\mu > 0$, we observe that

$$\begin{aligned} (1 - b^2 - c^2)^2 - 4b^2c^2 &= (1 - (b-c)^2)(1 - (b+c)^2) \\ &= \left[1 - \frac{|\lambda_1(x)|^{2(p-1)}}{\left[\phi_p(\mu,\lambda_1(x))\right]^{2(p-1)}}\right]\cdot\left[1 - \frac{|\lambda_2(x)|^{2(p-1)}}{\left[\phi_p(\mu,\lambda_2(x))\right]^{2(p-1)}}\right] \\ &> 0, \end{aligned}$$

where the inequality holds due to $|\lambda_i(x)| < \phi_p(\mu, \lambda_i(x))$ for $i = 1, 2$ and $\mu > 0$. With all of these, we see that the Schur complement of $1 - b^2 - c^2$ given as in (14) is a linear positive combination of the matrices $\left(I - \frac{x_2 x_2^T}{\|x_2\|^2}\right)$ and $\frac{x_2 x_2^T}{\|x_2\|^2}$, which yields that the Schur complement (14) of $1 - b^2 - c^2$ is positive semidefinite. Hence, the matrix $I - C^T C$ is also positive semidefinite, which is equivalent to saying $0 \le \lambda_{\min}(C^T C) \le \lambda_{\max}(C^T C) \le 1$. Thus, the proof is complete. $\square$

Theorem 4.1 indicates the Newton equation (11) in Algorithm 4.1 is solvable. It paves a way to show that the linear search (12) in Algorithm 4.1 is well-defined which is given in Theorem 4.2 as below. Indeed, the proof is very similar to the one in [17, Remark 2.1 (v)], we only present it here and omit its proof.

Theorem 4.2. *Suppose that Assumption 4.1 holds. Then, for $\triangle z \in \mathbb{R} \times \mathbb{R}^n$ given by (11), the linear search (12) is well-defined.*

Next, we discuss the convergence of Algorithm 4.1. To this end, we need the following results whose arguments are similar to the ones in [17, Remark 2.1].

Theorem 4.3. *Let H be defined as in (8). Suppose that Assumption 4.1 holds and that the sequence $\{z^k\}$ is generated by Algorithm 4.1. Then, the following results are hold.*

(a) *The sequences $\{\|H(z^k)\|\}$ and $\{\tau_k\}$ are monotonically non-increasing.*
(b) *$\beta\mu_k \geq \tau_k^2$ for all k.*
(c) *The sequence $\{\mu_k\}$ is monotonically non-increasing and $\mu_k > 0$ for all k.*
(d) *The sequence $\{z^k\}$ is bounded.*

Proof. (a) From definition of the line search in (12) and $\tau_k := \min\{1, \|H(z^k)\|\}$, it is clear that $\{\|H(z^k)\|\}$ and $\{\tau_k\}$ are monotonically non-increasing.

(b) We prove this conclusion by induction. First, by Algorithm 4.1, it is clear that $\tau_0^2 \leq \beta\mu_0$ with τ_0, β and μ_0 chosen in Algorithm 4.1. Secondly, we suppose that $\tau_k^2 \leq \beta\mu_k$ for some k. Then, for $k+1$, we have

$$
\begin{aligned}
\mu_{k+1} - \frac{\tau_{k+1}^2}{\beta} &= \mu_k + \alpha_k \triangle\mu_k - \frac{\tau_{k+1}^2}{\beta} \\
&= (1-\alpha_k)\mu_k + \alpha_k \frac{\tau_k^2}{\beta} - \frac{\tau_{k+1}^2}{\beta} \\
&\geq (1-\alpha_k)\frac{\tau_k^2}{\beta} + \alpha_k \frac{\tau_k^2}{\beta} - \frac{\tau_{k+1}^2}{\beta} \\
&\geq 0,
\end{aligned}
$$

where the second equality holds due to the Newton equation (11), and the second inequality holds due to part (a). Hence, it follows that $\beta\mu_k \geq \tau_k^2$ for all k.

(c) From the iterative scheme $z^{k+1} = z^k + \alpha_k \Delta z^k$, we know $\mu_{k+1} = \mu_k + \alpha_k \triangle\mu_k$. By the Newton equations (11) and the line search as in (12) again, it follows that

$$
\mu_{k+1} = (1-\alpha_k)\mu_k + \alpha_k \frac{\tau_k^2}{\beta} \geq (1-\alpha_k)\frac{\tau_k^2}{\beta} + \alpha_k \frac{\tau_k^2}{\beta} > 0
$$

for all k. On the other hand, we have

$$
\mu_{k+1} = (1-\alpha_k)\mu_k + \alpha_k \frac{\tau_k^2}{\beta} \leq (1-\alpha_k)\mu_k + \alpha_k\mu_k \leq \mu_k,
$$

where the first inequality holds due to part (b). Hence, the sequence $\{\mu_k\}$ is monotonically non-increasing and $\mu_k > 0$ for all k.

(d) From part (a), we know the sequence $\{\|H(z^k)\|\}$ is bounded. Thus, there is a constant C such that $\|H(z^k)\| \leq C$. In addition, since

$$
\begin{aligned}
& 4\left\|\lambda_1(x^k)u_x^{(1)} + \lambda_2(x^k)u_x^{(2)}\right\|^2 - \frac{\sqrt[p]{4}}{4}\left(|\lambda_1(x^k)| + |\lambda_2(x^k)|\right)^2 \\
&= \frac{1}{4}\left[(8 - 2\sqrt[p]{4})(|\lambda_1(x^k)|^2 + |\lambda_2(x^k)|^2) + \sqrt[p]{4}(|\lambda_1(x^k)| - |\lambda_2(x^k)|)^2\right] \\
&> 0 \ \ (\forall p > 1),
\end{aligned}
$$

it follows that

$$
\begin{aligned}
& \|H(z^k)\| \\
\geq & \left\| Ax^k + B\Phi_p(\mu_k, x^k) - b \right\| \\
\geq & \left\| Ax^k \right\| - \left\| B\Phi_p(\mu_k, x^k) \right\| - \|b\| \\
= & \sqrt{(x^k)^T A^T A x^k} - \sqrt{[\Phi_p(\mu_k, x^k)]^T B^T B \Phi_p(\mu_k, x^k)} - \|b\| \\
\geq & \sqrt{\lambda_{\min}(A^T A)}\|x^k\| - \sqrt{\lambda_{\max}(B^T B)\|\Phi_p(\mu_k, x^k)\|^2} - \|b\| \\
= & \sqrt{\lambda_{\min}(A^T A)}\|x^k\| - \sqrt{\lambda_{\max}(B^T B) \left\| \phi_p(\mu_k, \lambda_1(x^k))u_x^{(1)} + \phi_p(\mu_k, \lambda_2(x^k))u_x^{(2)} \right\|^2} - \|b\| \\
= & \sqrt{\lambda_{\min}(A^T A)}\|x^k\| - \sqrt{\lambda_{\max}(B^T B) \left[\phi_p^2(\mu_k, \lambda_1(x^k))\|u_x^{(1)}\|^2 + \phi_p^2(\mu_k, \lambda_2(x^k))\|u_x^{(2)}\|^2 \right]} - \|b\| \\
= & \sqrt{\lambda_{\min}(A^T A)}\|x^k\| - \sqrt{\lambda_{\max}(B^T B) \left[\frac{1}{2} \left(\sqrt[p]{(\mu_k^p + |\lambda_1(x^k)|^p)^2} + \sqrt[p]{(\mu_k^p + |\lambda_2(x^k)|^p)^2} \right) \right]} - \|b\| \\
\geq & \sqrt{\lambda_{\min}(A^T A)}\|x^k\| - \sqrt{\lambda_{\max}(B^T B)} \\
& \cdot \sqrt{\left[\frac{1}{2} \left((\mu_k^2 + |\lambda_1(x^k)|^2 + \sqrt[p]{2}\mu_k|\lambda_1(x^k)|) + (\mu_k^2 + |\lambda_2(x^k)|^2 + \sqrt[p]{2}\mu_k|\lambda_2(x^k)|) \right) \right]} - \|b\| \\
= & \sqrt{\lambda_{\min}(A^T A)}\|x^k\| \\
& - \sqrt{\lambda_{\max}(B^T B)}\sqrt{\mu_k^2 + \frac{1}{2}|\lambda_1(x^k)|^2 + \frac{1}{2}|\lambda_2(x^k)|^2 + \frac{\sqrt[p]{2}}{2}\mu_k(|\lambda_1(x^k)| + |\lambda_2(x^k)|)} - \|b\| \\
\geq & \sqrt{\lambda_{\min}(A^T A)}\|x^k\| \\
& - \sqrt{\lambda_{\max}(B^T B)}\sqrt{\mu_k^2 + \frac{1}{2}|\lambda_1(x^k)|^2 + \frac{1}{2}|\lambda_2(x^k)|^2 + 2\mu_k\|\lambda_1(x^k)u_x^{(1)} + \lambda_2(x^k)u_x^{(2)}\|} - \|b\| \\
= & \sqrt{\lambda_{\min}(A^T A)}\|x^k\| - \sqrt{\lambda_{\max}(B^T B)} \left[\mu_k + \|\lambda_1(x^k)u_x^{(1)} + \lambda_2(x^k)u_x^{(2)}\| \right] - \|b\| \\
= & \left(\sqrt{\lambda_{\min}(A^T A)} - \sqrt{\lambda_{\max}(B^T B)} \right) \|x^k\| - \sqrt{\lambda_{\max}(B^T B)}\mu_k - \|b\|.
\end{aligned}
$$

This together with $\|H(z^k)\| \leq C$ implies

$$
\|x^k\| \leq \frac{C + \sqrt{\lambda_{\max}(B^T B)}\mu_k + \|b\|}{\sqrt{\lambda_{\min}(A^T A)} - \sqrt{\lambda_{\max}(B^T B)}}
$$

holds for all k. Thus, the sequence $\{x^k\}$ is bounded. □

Theorem 4.4. *Suppose that Assumption 4.1 holds and that $\{z^k\}$ is generated by Algorithm 4.1. Then, any accumulation point of $\{z^k\}$ is a solution to the SOCAVE (2).*

Proof. From Theorem 4.3 (d), we know the sequence $\{z^k\}$ is bounded. Hence, there exists at least a accumulation point for the sequence $\{z^k\}$. Without loss of generality, let $\lim_{k\to\infty} z^k := z^\star = (\mu_\star, x^\star)$. Then, it follows that $H^\star := H(z^\star) = \lim_{k\to\infty} H(z^k)$ and $\tau_\star := \min\{1, \|H^\star\|\} = \lim_{k\to\infty} \min\{1, \|H(z^k)\|\}$. Now, we will show $H^\star = 0$. Suppose not, i.e., $\|H^\star\| > 0$. To proceed, we discuss two cases according to whether $\lim_{k\to\infty} \alpha_k = 0$ or $\alpha_k \geq \hat{\alpha} > 0$ with $\hat{\alpha} \in \mathbb{R}_{++}$.

Case 1: $\lim_{k\to\infty} \alpha_k = 0$. Then, from the line search (12), for the number $\overline{\alpha}_k := \frac{\alpha_k}{\delta}$ with all sufficiently large k, we have

$$
\|H(z_k + \overline{\alpha}_k \triangle z_k)\| > [1 - \sigma(1 - \frac{1}{\beta})\overline{\alpha}_k]\|H(z_k)\|.
$$

Furthermore, this leads to

$$
\frac{\|H(z^k + \overline{\alpha}_k \triangle z^k)\| - \|H(z^k)\|}{\overline{\alpha}_k} > -\sigma(1 - \frac{1}{\beta})\|H(z^k)\|. \tag{15}
$$

Besides, from Theorem 4.3 (c) again, we know $\mu^\star \geq 0$. It follows that the function H is continuously differentiable at the point $z^\star$. Taking $k \to \infty$ in the formula (15), we have

$$\frac{\langle H(z^\star), H'(z^\star)\triangle z^\star\rangle}{\|H(z^\star)\|} \geq -\sigma(1-\frac{1}{\beta})\|H(z^\star)\|. \tag{16}$$

This combining the Newton equations (11) yields

$$\begin{aligned}\frac{\langle H(z^\star), H'(z^\star)\triangle z^\star\rangle}{\|H(z^\star)\|} &= \frac{(\tau_\star)^2}{\beta\|H(z^\star)\|}\langle H(z^\star), e\rangle - \|H(z^\star)\| \\ &\leq \frac{(\tau_\star)^2\|H(z^\star)\|}{\beta\|H(z^\star)\|} - \|H(z^\star)\| \\ &\leq \frac{\tau_\star}{\beta} - \|H(z^\star)\| \\ &\leq (\frac{1}{\beta}-1)\|H(z^\star)\|, \end{aligned} \tag{17}$$

where the first inequality holds due to the Hölder inequality $\langle H(z^\star), e\rangle \leq \|H(z^\star)\|\|e\| = \|H(z^\star)\|$, the second and third inequality hold due to $\tau_\star = \min\{1, \|H(z^\star)\|\}$. Putting (16) and (17) together gives $\frac{1}{\beta} - 1 \geq -\sigma(1-\frac{1}{\beta})$. This contradicts $\sigma \in (0,1)$ and $\beta > 1$.

Case 2: $\alpha_k \geq \hat{\alpha} > 0$ for all k. From the line search (12), we have

$$\|H(z^{k+1})\| \leq \left[1-\sigma(1-\frac{1}{\beta})\hat{\alpha}\right]\|H(z^k)\| = \|H(z^k)\| - \sigma(1-\frac{1}{\beta})\hat{\alpha}\|H(z^k)\|.$$

Then, it follows from the boundedness of $\|H(z^k)\|$ that $\sum_{k=0}^{\infty}\hat{\alpha}\sigma(1-\frac{1}{\beta})\|H(z^k)\|$ is bounded. Moreover, we have $\lim_{k\to\infty}\|H(z^k)\| = 0$, i.e., $\|H^\star\| = 0$. This contradicts $\|H^\star\| > 0$.

Hence, from all the above, we show $H(z^\star) = 0$. That is, the element $x^\star$ is a solution of the SOCAVE (2). Then, the proof is complete. □

Now, we show the local quadratic convergence of Algorithm 4.1. In fact, we can achieve the following result by similar arguments as those in [37, Theorem 8]. For completeness, we also provide a detailed proof.

Theorem 4.5. *Let H be defined as in (8) and $z^\star$ be the unique solution to SOCAVE (2). Suppose that Assumption 4.1 holds and that all $V \in \partial H(z^\star)$ are nonsingular. Then, the whole sequence $\{z^k\}$ converges to $z^\star$, and $\|z^{k+1} - z^\star\| = O(\|z^k - z^\star\|^2)$.*

Proof. Since $z^\star$ is the solution to SOCAVE (2), using Assumption 4.1 and applying Theorem 4.1 yield that the Jacobian matrix $H'(z^k)$ is nonsingular for all z^k sufficiently close to $z^\star$. On the other hand, applying the condition that all $V \in \partial H(z^\star)$ are nonsingular and from [36, Proposition 3.1], we have $\|H'(z^k)^{-1}\| = O(1)$ for all z^k sufficiently close to $z^\star$. Because $z^\star$ is the solution to SOCAVE (2), it is clear that $z^\star$ is a solution of $H(z) = 0$. In addition, the function H is strongly semismooth, it follows that

$$\|H(z^k) - H(z^\star) - H'(z^k)(z^k - z^\star))\| = O(\|z^k - z^\star\|^2).$$

Thus, we have

$$\begin{aligned}\left\|z^k + \triangle z^k - z^\star\right\| &= \left\|z^k + H'(z^k)^{-1}\left[-H(z^k) + \frac{1}{\beta}\tau_k^2 e\right] - z^\star\right\| \\ &\leq \left\|H'(z^k)^{-1}\left(-H(z^k) + H'(z^k)(z^k - z^\star)\right)\right\| + \left\|H'(z^k)^{-1}\frac{1}{\beta}\tau_k^2 e\right\| \\ &\leq \left\|H'(z^k)^{-1}\left(-H(z^k) + H'(z^k)(z^k - z^\star)\right)\right\| + O(1)\left\|\frac{1}{\beta}\tau_k^2 e\right\| \\ &= O(\|H(z^k) - H(z^\star) - H'(z^k)(z^k - z^\star)\|) + O(\|H(z^k)\|^2) \\ &= O(\|z^k - z^\star\|^2) + O(\|z^k - z^\star\|^2) \\ &= O(\|z^k - z^\star\|^2)\end{aligned}$$

where the first equality holds due to the Newton equation (11), and the third equality holds since the function H is locally Lipschitz continuous near z^k. Then, the proof is complete. □

5. Numerical results

This section is devoted to the numerical results. First, we show the numerical comparison between the smoothing Newton algorithm and generalized Newton method. This provides the numerical evidence about why we adopt the smoothing Newton algorithm, not the generalized Newton algorithm, in this paper. Secondly, we use the performance profile to depict the comparison among different values of p. This shows that the smoothing Newton algorithm is not regularly affected when p is perturbed. Moreover, a suitable smoothing function from the class of smoothing functions is suggested in view of the numerical comparisons.

5.1. Smoothing Newton algorithm vs generalized Newton method

In this subsection, for fixed $p = 2$, we provide some numerical examples to evaluate the efficiency of Algorithm 4.1. In our tests, we choose parameters

$$\mu_0 = 0.1, \quad x_0 = \text{rand}(n, 1), \quad \delta = 0.5, \quad \sigma = 10^{-5} \quad \text{and} \quad \beta = \max(1, 1.01 * \tau_0^2/\mu).$$

We stop the iterations when $\|H(z_k)\| \le 10^{-6}$ or the number of iterations exceeds 100. All the experiments are done on a PC with Intel(R) CPU of 2.40 GHz and RAM of 4.00 GHz, and all the program codes are written in Matlab and run in Matlab environment. We consider the following four problems, and compute these problems by using Smoothing Newton Algorithm (SN for short) 4.1 and Generalized Newton method (GN for short) which introduced in [16], respectively. Illustrative examples further demonstrate the superiority of our proposed algorithm.

Problem 5.1. Consider the SOCAVE (2) which is generated in the following way: first choose two random matrices $B, C \in \mathbb{R}^{n\times n}$ from a uniformly distribution on $[-10, 10]$ for every element. We compute the maximal singular value σ_1 of B and the minimal singular value σ_2 of C, and let $\sigma := \min\{1, \sigma_2/\sigma_1\}$. Next, we divide C by σ multiplied by a random number in the interval $[0, 1]$, and the resulting matrix is denoted as A. Accordingly, the minimum singular values of A exceeds the maximal singular value of B. We choose randomly $b \in \mathbb{R}^n$ on $[0, 1]$ for every element. By Algorithm 4.1 in this paper, the resulting SOCAVE (2) is solvable. The initial point is chosen in the range $[0, 1]$ entry-wisely. Note that a similar way to construct the problem was given in [16].

Problem 5.2. Consider the SOCAVE (2) which is generated in the following way: choose two random matrices $C, D \in \mathbb{R}^{n\times n}$ from a uniformly distribution on $[-10, 10]$ for every element, and compute their singular value decompositions $C := U_1 S_1 V_1^T$ and $D := U_2 S_2 V_2^T$ with diagonal matrices S_1 and S_2; unitary matrices U_1, V_1, U_2 and V_2. Then, we choose randomly $b, c \in \mathbb{R}^n$ on $[0, 10]$ for every element. Next, we take $a \in \mathbb{R}^n$ by setting $a_i = c_i + 10$ for all $i \in \{1, \ldots, n\}$, so that $a \ge b$. Set $A := U_1 \text{Diag(a)} V_1^T$ and $B := U_2 \text{Diag(b)} V_2^T$, where Diag(x) denotes a diagonal matrix with its i-th diagonal element being x_i. The gap between the minimal singular value of A and the maximal singular value of B is limited and can be very small. We choose randomly $b \in \mathbb{R}^n$ in $[0, 10]$. The initial point is chosen in the range $[0, 1]$ entry-wisely.

Problem 5.3. Consider the SOCAVE (2) which is generated in the following way: choose two random matrices $A, B \in \mathbb{R}^{n\times n}$ from a uniformly distribution on $[-10, 10]$ for every element. In order to ensure that the SOCAVE (2) is solvable, we update the matrix A by the following: let $[U S V] = svd(A)$. If $\min\{S(i,i)\} = 0$ for $i = 0, 1, \cdots, n$, we make $A = U(S + 0.01E)V$, and then $A = \frac{\lambda_{max}(B^T B) + 0.01}{\lambda_{min}(A^T A)} A$. We choose randomly $b \in \mathbb{R}^n$ on $[0, 10]$ for every element. The initial point is chosen in the range $[0, 1]$ entry-wisely.

Problem 5.4. We consider the SOCAVE (2) which is generated the same as Problem 5.1. But, here the SOC is given by $\mathcal{K} := \mathcal{K}^{n_1} \times \cdots \times \mathcal{K}^{n_r}$, where $n_1 = \cdots = n_r = \frac{n}{r}$.

The above Problems 5.1–5.4 are both generated randomly. Below, as suggested by the reviewer, we consider a real application problem. It is well known that the second-order cone linear complementarity problem (SOCLCP) has various applications in engineering, control, finance, robust optimization and combinatorial optimization since the KKT system of a second-order cone programming can be recast an SOCLCP. In general, the SOCLCP is to find $x, y \in R^n$ such that

$$Mx + Py = c, \quad x \in \mathcal{K}, \quad y \in \mathcal{K}, \quad x^{\mathrm{T}} y = 0, \tag{18}$$

where $M, P \in R^{n\times m}$ are given matrices and $c \in R^n$ is given vector. From [16, Theorem 1.1], we know that the SOCLCP (18) is equivalent to the SOCAVE (2). In view of this, the next experiment is on this case.

Problem 5.5. Consider the SOCLCP with $P = -I$, which is generated in the following way: First, we generate a matrix B and a vector b as those given in Problem 5.1. Then, let d be a random number in $[0, 1]$. We set $M := BB^{\mathrm{T}} + (1 + d)I$ and $c := 0.5(M(b + |b|) + |b| - b)$ to ensure the solvability of the SOCLCP. We test the above SOCLCP by casting it into an SOCAVE according to [16, Theorem 1.1], i.e., we implement the corresponding SOCAVE with $A = M + I$, $B = M - I$ and $b = 2c$. Moreover, the initial point is chosen in the range $[0, 1]$ entry-wisely.

Table 1
Numerical results for Problem 5.1.

SN							GN					
n	Ares	Itn	Time	Maxi	Mini	Fails	Ares	Itn	Time	Maxi	Mini	Fails
100	8.618e−08	2.8	0.078	3	2	0	9.992e−08	2.8	0.349	3	2	0
200	4.901e−08	2.6	0.051	3	2	0	6.904e−10	2.9	0.134	3	2	0
300	1.574e−08	2.7	0.122	3	2	0	3.779e−09	2.9	0.231	3	2	0
400	3.041e−09	2.7	0.232	3	2	0	9.155e−08	2.7	0.326	3	2	0
500	1.778e−07	2.2	0.300	3	2	0	1.445e−07	2.6	0.421	3	2	0
600	1.385e−07	2.5	0.498	3	2	0	5.626e−08	2.8	0.844	3	2	0
700	2.578e−07	2.4	0.668	3	2	0	1.527e−08	2.6	1.334	3	2	0
800	2.356e−07	2.1	0.771	3	2	0	6.846e−08	2.6	1.905	3	2	0
900	2.420e−08	2.5	1.031	3	2	0	1.272e−09	2.7	2.685	3	2	0
1000	4.718e−08	2.5	1.193	3	2	0	1.135e−07	2.7	3.691	3	2	0
1500	2.027e−07	2.3	1.919	3	2	0	6.417e−08	2.6	13.369	3	2	0
2000	3.121e−08	2.2	3.892	3	2	0	1.015e−07	2.5	32.982	3	2	0
2500	1.565e−07	2.1	6.625	3	2	0	3.940e−08	2.5	53.510	3	2	0
3000	1.028e−07	2.3	12.340	3	2	0	1.293e−07	2.5	87.910	3	2	0

Table 2
Numerical results for Problem 5.2.

SN							GN					
n	Ares	Itn	Time	Maxi	Mini	Fails	Ares	Itn	Time	Maxi	Mini	Fails
100	2.884e−07	4.2	0.050	5	4	0	1.920e−07	4.4	0.134	5	4	0
200	4.556e−07	4.3	0.067	5	4	0	2.637e−07	4.6	0.346	5	4	0
300	2.805e−07	4.5	0.172	5	4	0	3.522e−07	4.4	0.615	5	4	0
400	2.453e−07	4.6	0.312	5	4	0	2.617e−07	4.6	0.863	5	4	0
500	1.809e−13	5.0	0.516	5	5	0	1.037e−07	4.8	1.440	5	4	0
600	1.870e−07	4.8	0.680	5	4	0	3.414e−12	5.0	2.346	5	5	0
700	2.550e−13	5.0	0.880	5	5	0	6.571e−08	4.9	3.535	5	4	0
800	2.868e−13	5.0	1.083	5	5	0	1.606e−07	4.8	5.317	5	4	0
900	7.559e−08	4.9	1.201	5	4	0	2.485e−07	4.7	7.596	5	4	0
1000	3.595e−13	5.0	1.572	5	5	0	1.662e−07	4.8	10.552	5	4	0
1500	5.412e−13	5.0	4.196	5	5	0	1.782e−11	5.0	34.400	5	5	0
2000	7.230e−13	5.0	8.962	5	5	0	2.851e−11	5.0	79.108	5	5	0
2500	8.893e−13	5.0	17.207	5	5	0	4.451e−11	5.0	146.769	5	5	0
3000	1.054e−12	5.0	29.175	5	5	0	6.119e−11	5.0	247.029	5	5	0

Table 3
Numerical results for Problem 5.3.

SN							GN					
n	Ares	Itn	Time	Maxi	Mini	Fails	Ares	Itn	Time	Maxi	Mini	Fails
100	7.928e−10	3.0	0.048	3	3	0	2.085e−08	3.0	0.075	3	3	0
200	9.461e−10	3.0	0.062	3	3	0	4.297e−09	3.0	0.108	3	3	0
300	2.388e−10	3.0	0.122	3	3	0	5.843e−08	2.9	0.237	3	2	0
400	5.780e−11	3.0	0.236	3	3	0	3.841e−08	2.8	0.379	3	2	0
500	1.133e−08	2.9	0.360	3	2	0	1.183e−09	2.9	0.501	3	2	0
600	2.655e−08	2.9	0.566	3	2	0	1.225e−10	3.0	0.627	3	3	0
700	2.202e−11	3.0	0.807	3	3	0	2.525e−10	3.0	0.978	3	3	0
800	8.893e−08	2.8	0.975	3	2	0	2.563e−10	3.0	1.576	3	3	0
900	1.818e−08	2.9	1.240	3	2	0	2.505e−10	3.0	2.374	3	3	0
1000	6.951e−10	3.0	1.502	3	3	0	3.247e−10	3.0	3.367	3	3	0
1500	4.225e−08	2.9	2.482	3	2	0	4.245e−10	3.0	11.625	3	3	0
2000	6.979e−08	2.6	4.683	3	2	0	1.705e−09	3.0	27.704	3	3	0
2500	9.459e−10	2.9	9.441	3	2	0	1.376e−09	3.0	53.306	3	3	0
3000	5.624e−08	2.9	15.765	3	2	0	1.943e−08	2.8	91.226	3	2	0

In our experiments, every set of the simulations for every problem is randomly generated ten times, and the numerical results are listed in Tables 1–5, respectively. In Tables 1–5, n denotes the size of testing problem; *ares* denotes the average value of $\|H(z^k)\|$ when the test stops; *itn* denotes the average value of the iteration numbers; *time* denotes the average value of the CPU time in seconds; *maxit* and *minit* denote the maximal value and the minimal value of the iteration numbers, respectively; and *fails* denotes that the times of test is failed. From the numerical results that are presented in Tables 1–5, it is easy to see that the proposed smoothing Newton method is effective for solving all the simulated SOCAVE problems. For the SOCLCP, although the smoothing Newton method performs slightly less than the generalized Newton method, the difference is marginal. To sum up, both approaches are competitive and can be employed to solve SOCAVE.

Table 4
Numerical results for Problem 5.4.

SN								GN					
n	r	Ares	Itn	Time	Maxit	Minit	Fails	Ares	Itn	Time	Maxit	Minit	Fails
	2	9.933e−08	2.4	1.318	3	2	0	2.995e−10	2.9	3.627	3	2	0
	4	1.174e−07	2.5	1.245	3	2	0	1.594e−08	2.6	3.106	3	2	0
1000	5	1.056e−07	2.4	1.293	3	2	0	9.657e−08	2.7	3.115	3	2	0
	10	3.380e−13	5.0	1.791	5	5	0	3.971e−08	2.5	3.218	3	2	0
	20	3.360e−13	5.0	2.103	5	5	0	5.291e−08	2.7	3.181	3	2	0
	2	1.971e−08	2.6	5.084	3	2	0	2.494e−08	2.6	28.888	3	2	0
	4	1.047e−07	2.3	4.270	3	2	0	5.363e−08	2.6	29.002	3	2	0
2000	5	1.257e−07	2.5	4.813	3	2	0	1.360e−08	2.8	29.055	3	2	0
	10	6.689e−13	5.0	10.463	5	5	0	1.360e−08	2.8	29.055	3	3	0
	20	6.653e−13	5.0	11.255	5	5	0	1.360e−08	2.8	29.055	3	4	0
	2	1.560e−07	2.1	12.312	3	2	0	2.496e−07	2.5	90.699	3	2	0
	4	1.162e−07	2.5	14.457	3	2	0	1.609e−07	2.3	89.813	3	2	0
3000	5	3.156e−07	2.2	12.995	3	2	0	6.872e−0	2.4	88.921	3	2	0
	10	9.922e−13	5.0	32.011	5	5	0	1.688e−07	2.4	90.041	3	2	0
	20	1.016e−12	5.0	33.877	5	5	0	1.411e−08	2.5	88.949	3	2	0

Table 5
Numerical results for Problem 5.5.

SN							GN					
n	Ares	Itn	Time	Maxi	Mini	Fails	Ares	Itn	Time	Maxi	Mini	Fails
100	1.159e−10	3.0	0.011	3	3	0	6.353e−11	2.2	0.019	3	2	0
200	3.093e−10	3.0	0.017	3	3	0	2.691e−10	2.0	0.009	2	2	0
300	5.937e−10	3.0	0.039	3	3	0	4.905e−10	2.1	0.025	3	2	0
400	1.162e−09	3.0	0.100	3	3	0	1.205e−09	2.1	0.064	3	2	0
500	1.755e−09	3.0	0.129	3	3	0	1.538e−09	2.1	0.112	3	2	0
600	2.439e−09	3.0	0.223	3	3	0	2.239e−09	2.0	0.154	2	2	0
700	3.897e−09	3.0	0.298	3	3	0	3.522e−09	2.0	0.229	2	2	0
800	4.918e−09	3.0	0.399	3	3	0	3.407e−09	2.0	0.289	2	2	0
900	5.487e−09	3.0	0.566	3	3	0	5.359e−09	2.0	0.643	2	2	0
1000	7.328e−09	3.0	0.971	3	3	0	5.994e−09	2.0	0.530	2	2	0
1500	1.687e−08	3.0	2.140	3	3	0	1.354e−08	2.0	1.635	2	2	0
2000	3.121e−08	3.0	4.733	3	3	0	2.519e−08	2.0	3.597	2	2	0
2500	4.956e−08	3.0	8.784	3	3	0	4.010e−08	2.0	6.387	2	2	0
3000	6.581e−08	3.0	14.508	3	3	0	6.062e−08	2.0	10.855	2	2	0

5.2. Numerical comparisons with different values of p

In this subsection, we observe the numerical comparison of Algorithm 4.1 with different values of p. In particular, we consider the performance profile which is introduced in [44] as a means. In other words, we regard Algorithm 4.1 corresponding to different $p = 1.1, 2, 3, 10, 20, 80$ as a solver, and assume that there are n_s solvers and n_q test problems from the test set $\mathcal{P}$ which is generated randomly. We are interested in using the computing time as performance measure for Algorithm 4.1 with different p. For each problem q and solver s, let

$$f_{q,s} = \text{computing time required to solve problem } q \text{ by solver } s.$$

We employ the performance ratio

$$r_{q,s} := \frac{f_{q,s}}{\min\{f_{q,s} : s \in \mathcal{S}\}},$$

where $\mathcal{S}$ is the six solvers set. We assume that a parameter $r_{q,s} \le r_M$ for all q, s are chosen, and $r_{q,s} = r_M$ if and only if solver s does not solve problem q. In order to obtain an overall assessment for each solver, we define

$$\rho_s(\tau) := \frac{1}{n_q}\text{size}\{q \in \mathcal{P} : r_{q,s} \le \tau\},$$

which is called the performance profile of the computing time for solver s. Then, $\rho_s(\tau)$ is the probability for solver $s \in \mathcal{S}$ that a performance ratio $f_{q,s}$ is within a factor $\tau \in \mathbb{R}$ of the best possible ratio.

Fig. 1 depicts the performance profile of computation time in Algorithm 4.1 in the range of $\tau \in [1, 1.8]$ for six solvers on 200 test problem which are generated randomly from Problem 5.1 to Problem 5.4. The six solvers correspond to Algorithm 4.1 with $p = 1.1, 2, 3, 10, 20, 80$, respectively. From this figure, we see that the algorithm is not regularly affected

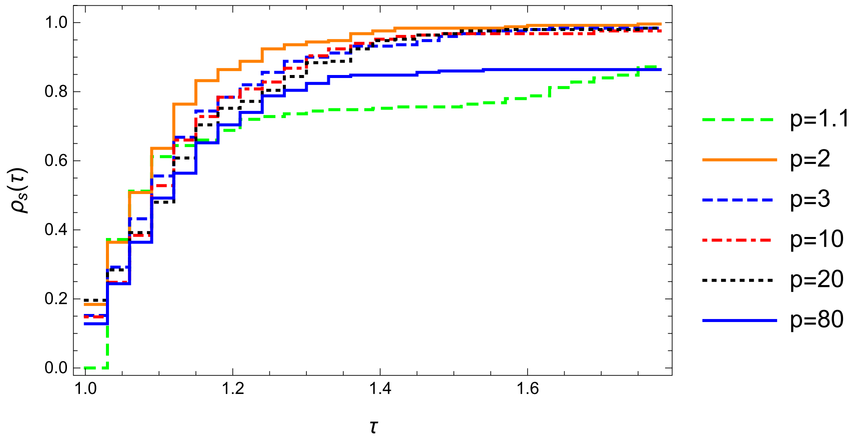

Fig. 1. Performance profile of computation time of Algorithm 4.1 with different p.

when p is perturbed. Moreover, we observe that the large value of p and the small value of p which is close to 1 seem not the good choices of being employed to work with the proposed algorithm. When taking $p = 2$ along with Algorithm 4.1, it has best numerical performance. This suggests that ϕ_2 is the best choice of function to be applied in the smoothing Newton method.

6. Concluding remarks

In this paper, we have studied the absolute value equation associated with SOC, which is a natural extension of the standard absolute value equation. Based on a class of smoothing functions, the smoothing Newton algorithm is proposed to solve SOCAVE iteratively. The algorithm is shown to be well-defined, quadratically convergent under suitable conditions. Some preliminary numerical results are reported which explain the efficiency of the proposed method. Although, the main idea for proving the convergence is similar to the one used in the literature, the analysis is indeed more subtle and involves more techniques due to the feature of SOC. Moreover, two kinds of numerical comparisons are presented in this paper. The first one is the numerical comparison between the smoothing Newton algorithm and generalized Newton method. This provides numerical evidence of indicating that the smoothing Newton algorithm and the generalized Newton algorithm are competitive for solving SOCAVE. Another comparison is based on various values of p, from which we see that the large value of p and the small value of p which is close to 1 are not suitable to work with the proposed algorithm. In particular, when taking $p = 2$ along with Algorithm 4.1, the numerical performance is the best. This suggests that ϕ_2 is the best choice from the class of smoothing functions to be applied in the smoothing Newton method for solving SOCAVE.

References

[1] L. Caccetta, B. Qu, G.-L. Zhou, A globally and quadratically convergent method for absolute value equations, Comput. Optim. Appl. 48 (2011) 45–58.
[2] J.-S. Chen, X. Chen, P. Tseng, Analysis of nonsmooth vector-valued functions associated with second-order cones, Math. Program. 101 (1) (2004) 95–117.
[3] J.-S. Chen, P. Tseng, An unconstrained smooth minimization reformulation of second-order cone complementarity problem, Math. Program. 104 (2005) 293–327.
[4] J.-S. Chen, The convex and monotone functions associated with second-order cone, Optimization 55 (4) (2006) 363–385.
[5] J.-S. Chen, The semismooth-related properties of a merit function and a descent method for the nonlinear complementarity problem, J. Glob. Optim. 36 (2006) 565–580.
[6] J.-S. Chen, On some NCP-functions based on the generalized Fischer–Burmeister function, Asia-Pac. J. Oper. Res. 24 (2007) 401–420.
[7] J.-S. Chen, S.-H. Pan, A family of NCP-functions and a descent method for the nonlinear complementarity problem, Comput. Optim. Appl. 40 (2008) 389–404.
[8] J.-S. Chen, S.-H. Pan, T.-C. Lin, A smoothing Newton method based on the generalized Fischer–Burmeister function for MCPs, Nonlinear Anal. 72 (2010) 3739–3758.
[9] J.-S. Chen, Z.-H. Huang, C.-Y. She, A new class of penalized NCP-functions and its properties, Comput. Optim. Appl. 50 (2011) 49–73.
[10] J.-S. Chen, S.-H. Pan, A survey on SOC complementarity functions and solution methods for SOCPs and SOCCPs, Pac. J. Optim. 8 (1) (2012) 33–74.
[11] U. Faraut, A. Koranyi, Analysis on Symmetric Cones, Oxford Mathematical Monographs, Oxford University Press, New York, 1994.
[12] M. Fukushima, Z.-Q. Luo, P. Tseng, Smoothing functions for second-order-cone complementarity problems, SIAM J. Optim. 12 (2002) 436–460.
[13] F. Facchinei, J. Pang, Finite-Dimensional Variational Inequalities and Complementarity Problems, Springer, New York, 2003.
[14] S.-L. Hu, Z.-H. Huang, A note on absolute value equations, Optim. Lett. 4 (2010) 417–424.
[15] S.-L. Hu, Z.-H. Huang, J.-S. Chen, Properties of a family of generalized NCP-functions and a derivative free algorithm for complementarity problems, J. Comput. Appl. Math. 230 (2009) 69–82.
[16] S.-L. Hu, Z.-H. Huang, Q. Zhang, A generalized Newton method for absolute value equations associated with second order cones, J. Comput. Appl. Math. 235 (2011) 1490–1501.
[17] Z.-H. Huang, Y. Zhang, W. Wu, A smoothing-type algorithm for solving system of inequalities, J. Comput. Appl. Math. 220 (2008) 355–363.
[18] R.A. Horn, C.R. Johnson, Matrix Analysis, Cambridge University Press, Cambridge, 1985.
[19] Z.-H. Huang, T. Ni, Smoothing algorithms for complementarity problems over symmetric cones, Comput. Optim. Appl. 45 (2010) 557–579.

[20] Z.-H. Huang, L. Qi, D.-F. Sun, Sub-quadratic convergence of a smoothing Newton algorithm for the P_0- and monotone LCP, Math. Program. 99 (2010) 423–441.
[21] X.-Q. Jiang, A smoothing Newton method for solving absolute value equations, Adv. Mater. Res. 765–767 (2013) 703–708.
[22] X.-Q. Jiang, Y. Zhang, A smoothing-type algorithm for absolute value equations, J. Ind. Manag. Optim. 9 (2013) 789–798.
[23] S. Ketabchi, H. Moosaei, Minimum norm solution to the absolute value equation in the convex case, J. Optim. Theory Appl. 154 (2012) 1080–1087.
[24] L. Kong, J. Sun, N. Xiu, A regularized smoothing Newton method for symmetric cone complementarity problems, SIAM J. Optim. 19 (2008) 1028–1047.
[25] O.L. Mangasarian, Absolute value programming, Comput. Optim. Appl. 36 (2007) 43–53.
[26] O.L. Mangasarian, Absolute value equation solution via concave minimization, Optim. Lett. 1 (2007) 3–5.
[27] O.L. Mangasarian, A generalized Newton method for absolute value equations, Optim. Lett. 3 (2009) 101–108.
[28] O.L. Mangasarian, Knapsack feasibility as an absolute value equation solvable by successive linear programming, Optim. Lett. 3 (2009) 161–170.
[29] O.L. Mangasarian, Primal-dual bilinear programming solution of the absolute value equation, Optim. Lett. 6 (2012) 1527–1533.
[30] O.L. Mangasarian, Absolute value equation solution via dual complementarity, Optim. Lett. 7 (2013) 625–630.
[31] O.L. Mangasarian, Linear complementarity as absolute value equation solution, Optim. Lett. 8 (2014) 1529–1534.
[32] O.L. Mangasarian, Absolute value equation solution via linear programming, J. Optim. Theory Appl. 161 (2014) 870–876.
[33] X.-H. Miao, J.-S. Chen, The solvability and unique solvability of the absolute value equations associated with second-order cone, 2016, submitted for publication.
[34] O.L. Mangasarian, R.R. Meyer, Absolute value equation, Linear Algebra Appl. 419 (2006) 359–367.
[35] O.A. Prokopyev, On equivalent reformulations for absolute value equations, Comput. Optim. Appl. 44 (2009) 363–372.
[36] L. Qi, J. Sun, A nonsmooth version of Newton's method, Math. Program. 58 (1993) 353–367.
[37] L. Qi, D. Sun, G.L. Zhou, A new look at smoothing Newton methods for nonlinear complementarity problems and box constrained variational inequality problems, Math. Program. 87 (2000) 1–35.
[38] J. Rohn, A theorem of the alternative for the equation $Ax + B|x| = b$, Linear Multilinear Algebra 52 (2004) 421–426.
[39] J. Rohn, Solvability of systems of interval linear equations and inequalities, in: M. Fiedler, J. Nedoma, J. Ramik, J. Rohn, K. Zimmermann (Eds.), Linear Optimization Problems with Inexact Data, Springer, 2006, pp. 35–77.
[40] J. Rohn, An algorithm for solving the absolute value equation, Electron. J. Linear Algebra 18 (2009) 589–599.
[41] S. Yamanaka, M. Fukushima, A branch and bound method for the absolute value programs, Optimization 63 (2014) 305–319.
[42] Y. Zhang, Z.-H. Huang, A nonmonotone smoothing-type algorithm for solving a system of equalities and inequalities, J. Comput. Appl. Math. 233 (2010) 2312–2321.
[43] C. Zhang, Q.-J. Wei, Global and finite convergence of a generalized Newton method for absolute value equations, J. Optim. Theory Appl. 143 (2009) 391–403.
[44] E.D. Dolan, J.J. More, Benchmarking optimization software with performance profiles, Math. Program. 91 (2002) 201–213.