NUMERICAL ALGEBRA,
CONTROL AND OPTIMIZATION
Volume **12**, Number **1**, March **2022**

doi:10.3934/naco.2021050

pp. **47–61**

LEVENBERG-MARQUARDT METHOD FOR ABSOLUTE VALUE EQUATION ASSOCIATED WITH SECOND-ORDER CONE

Xin-He Miao, Kai Yao

School of Mathematics, Tianjin University
Tianjin 300072, P.R. China

Ching-Yu Yang and Jein-Shan Chen*

Department of Mathematics
National Taiwan Normal University, Taipei 11677, Taiwan

Abstract. In this paper, we suggest the Levenberg-Marquardt method with Armijo line search for solving absolute value equations associated with the 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. We analyze the convergence of the proposed algorithm. For numerical reports, we not only show the efficiency of the proposed method, but also present numerical comparison with smoothing Newton method. It indicates that the proposed algorithm could also be a good choice for solving the SOCAVE.

1. **Introduction.** The standard absolute value equation (AVE for short) currently being studied extensively is the main focus in this paper. For the AVE, there are two types of them. The first type is in the form of

$$Ax - |x| = b. \tag{1}$$

Another one is a more general AVE, which is in the form of

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

where $A \in \mathbb{R}^{n\times n}$, $0 \neq B \in \mathbb{R}^{n\times n}$ and $b \in \mathbb{R}^n$. Here $|x|$ denotes the componentwise absolute value of vector $x \in \mathbb{R}^n$, i.e., $|x| = (|x_1|, |x_2|, \cdots, |x_n|)^T \in \mathbb{R}^n$. In fact, when B is nonsingular, the AVE (2) reduces to the AVE (1). In particular, the AVE (2) is just the AVE (1) if $B = -I$, where I is the identity matrix.

It is known that the AVE (1) was first introduced by Rohn in [30] in 2004 and are capable to formulate many optimization problems [18, 21, 26, 29]. Due to this, the AVE has attracted much attention recently. For standard absolute value equation, there are two main research directions for it. One is on the theoretical side, while the other one focuses on the algorithm for solving the absolute value equation. For the theoretical side, Mangasarian and Meyer [24] show that the AVE (1) is equivalent to

2020 *Mathematics Subject Classification.* Primary: 26B05, 65K10; Secondary: 26B35, 90C33.
Key words and phrases. Second-order cone, Absolute value equations, Levenberg-Marquardt algorithm, Armijo line search.
The first author is supported by by National Natural Science Foundation of China (No. 11471241) and Natural Science Foundation of Inner Mongolia (No. 2019LH01001). The last author is supported by Ministry of Science and Technology, Taiwan.

* Corresponding author: Jein-Shan Chen.

the bilinear program, the generalized LCP (linear complementarity problem), and the standard LCP provided that 1 is not an eigenvalue of A. In the setting of the second-order cone, Hu, Huang and Zhang [12] have shown that the absolute value equation associated with the second-order cone (SOCAVE) (2) is equivalent to the below problem: 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 $c \in \mathbb{R}^n$. Note that the above problem is not a standard second-order cone linear complementarity problem (SOCLCP) because there exists one additional equation $Mx + Py = c$ therein. In light of this, Miao et.al. [27] have shown that the SOCAVE (2) can be further converted into a standard SOCLCP. In addition, about the study of the properties of solutions for the AVE (2), under the condition of solvability, the AVE (2) can have either unique solution or multiple (e.g., exponentially many) solutions. Moreover, various sufficient conditions on solvability and non-solvability of the AVE (1) and (2) with unique and multiple solutions are discussed in [24, 29]. There are many other theoretical results, see [11, 16, 18, 19, 22, 26, 24, 29, 30]. Towards to solutions of the AVE, various numerical methods for solving the AVE were proposed, for example, Mangasarian [20], Zhang and Wei [35] all considered a generalized Newton method for the AVE. Yamanaka and Fukushima [32] proposed a branch and bound method for the absolute value programs. There are also other many numerical methods for solving the standard AVE (2) in the literature, see [2, 14, 15, 20, 21, 23, 31, 35] and references therein.

In this paper, we focus on the type of absolute value equation associated with the second-order cone, denoted by the SOCAVE, whose format looks like the AVE (1):

$$Ax - |x| = b, \tag{3}$$

where $A \in \mathbb{R}^{n\times n}$ and $b \in \mathbb{R}^n$ are the same as those in (1). However, $|x|$ in the SOCAVE (3) denotes the square root of the Jordan product "$\circ$" of x and x associated with the second-order cone (SOC), that is, $|x| := \sqrt{x^2} = \sqrt{x \circ x}$. In fact, by applying the SOC-function, Jordan product [4] and their related properties, there has explicit expression for $|x| \in \mathcal{K}$ in the SOCAVE (3), where $\mathcal{K}$ denotes the general second-order cone. More details about the second-order cone, Jordan product and $|x|$ will be introduced in the next section.

As mentioned above, several mathematical problems including linear programming, bimatrix games can be reduced to the system of absolute value equations. Accordingly, we see that the SOCAVE (3) play the same role in various optimizations involving the second-order cones since the SOCAVE is equivalent to the SOCLCP under the suitable conditions. The SOCLCP has various applications in engineering, control, finance, robust optimization and combinatorial optimization. In this paper, we are interested in Levenberg-Marquardt method for solving the SOCAVE (3). The Levenberg-Marquardt method [10, 13, 17, 25, 33, 34] was proposed for solving nonlinear problems $F(x) = 0$. This method can be viewed as a combination of the steepest descent and Gauss-Newton method. The classical Levenberg-Marquardt method computes the search direction d_k by

$$d_k = -(J(x_k)^T J(x_k) + \mu_k I)^{-1} J(x_k)^T F(x_k), \tag{4}$$

where $J(x_k) = F'(x_k)$ denotes the Jacobian matrix of the function F at x_k, $\mu_k > 0$ is a parameter and I is the identity matrix. As stated in [13], the direction d_k in

(4) is unique, which is a nice feature of Levenberg-Marquardt method compared to the Newton method or the Gauss-Newton method. In our setting, $F(x)$ is defined by

$$F(x) = Ax - |x| - b.$$

In this paper, we consider the Levenberg-Marquardt method with Armijo line search. Besides analyzing the convergence of the proposed method, we also present some preliminary numerical results to illustrate the efficiency of the proposed method. On the other hand, we also compare with the smoothing Newton method, which is a promising way for solving the SOCAVE (3) [27, 28]. The numerical experiments indicate that the Levenberg-Marquardt method is competitive on some certain problems. This suggests that not only is the smoothing Newton method good for solving the SOCAVE (3), but the Levenberg-Marquardt method is also another good choice.

There are a few words about our notations in this paper. All vectors are column vectors unless transposed to a row vector by a prime T. $\mathbb{R}^n$ denotes the space of n-dimensional real space. $x^T y$ denotes the inner product of two vectors x and y in the n-dimensional real space. $\mathbb{R}_+$ and $\mathbb{R}_{++}$ denote the nonnegative and positive reals. For $x \in \mathbb{R}^n$, the 2-norm (i.e., Euclidean norm) is denoted by $\|x\|$. A matrix A^T will denote the transpose of A, the identity matrix of arbitrary dimension is denoted by I.

The remaining parts of this paper are organized as follows. In section 2, we recall some basic concepts and background knowledge about second-order cone, the projection onto the SOC and the expression of the absolute value function associated with the SOC. Besides, we also define a vector-valued smoothing function Ψ based on the SOC. In Section 3, we propose the Levenberg-Marquardt method with Armijo line search for solving the SOCAVE (3), and discuss the convergence of the proposed method. In Section 4, simulations, numerical results, and numerical comparison are presented.

2. **Preliminaries.**

In this section, we review some basic concepts and useful results regarding second-order cone and Jordan product, which will be extensively used in the subsequent analysis. For a comprehensive treatment of second-order cone, more details can be found in [3, 4, 5, 6, 7, 9].

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

$$\mathcal{K}^n := \{(x_1, x_2) \in \mathbb{R} \times \mathbb{R}^{n-1} \big| \|x_2\| \leq x_1\},$$

where $\|\cdot\|$ denotes the Euclidean norm. It is well known that the second-order cone is a special case of symmetric cones [9]. Besides, we can see that for $n = 1$, the second-order cone $\mathcal{K}^n$ is the set of nonnegative reals $\mathbb{R}_+$. In general, the general second-order cone $\mathcal{K}$ is the Cartesian product of the SOCs, i.e.,

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

Since all the analysis can be carried out on the setting of Cartesian product, without loss of generality, we focus on the single second-order cone $\mathcal{K}^n$ in this paper.

Next, we will introduce the concept of Jordan product associated with SOC $\mathcal{K}^n$. 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 by

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

Unlike scalar or matrix multiplication, the Jordan product is not associative, which is a main source of complication in the analysis of optimization problems involved SOC, see [7, 9] and references therein. Moreover, the identity element under the Jordan product is $e = (1, 0, \cdots, 0)^T \in \mathbb{R}^n$. Based on these definitions, x^2 means the Jordan product of x and x, i.e., $x^2 = x \circ x$; and $\sqrt{x}$ with $x \in \mathcal{K}^n$ denotes the vector satisfying $\sqrt{x} \circ \sqrt{x} = x$. On the basis of these concepts, the vector $|x|$ in (3) is computed by

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

In order to write out the expression of $|x|$ easily and explicitly, we first recall the spectral decomposition of x with respect to SOC. For any $x = (x_1, x_2) \in \mathbb{R} \times \mathbb{R}^{n-1}$, the *spectral decomposition* of x with respect to SOC [4, 6, 7] is given by

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

where $\lambda_1(x)$ and $\lambda_2(x)$ are the spectral values of x with $\lambda_i(x) = x_1 + (-1)^i \|x_2\|$ for $i = 1, 2$, while $u_x^{(1)}$ and $u_x^{(2)}$ are the corresponding spectral vectors of x given by

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

with $\nu \in \mathbb{R}^{n-1}$ being any vector satisfying $\|\nu\| = 1$ for $i = 1, 2$. It is easy to find that the spectral decomposition of $x \in \mathbb{R}^n$ is unique if $\|x_2\| \neq 0$.

In the next content of this section, we recall the projection onto $\mathcal{K}^n$. We let x_+ be the projection of x onto $\mathcal{K}^n$, and x_- denote the projection of $-x$ onto its dual cone of $\mathcal{K}^n$, where the dual cone is defined as

$$(\mathcal{K}^n)^* := \{y \in \mathbb{R}^n \mid \langle x, y \rangle \geq 0, \quad \forall x \in \mathcal{K}^n\}.$$

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 second-order cone, 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 [7, 9] as below:

$$x_+ = \begin{cases} x & \text{if } x \in \mathcal{K}^n, \\ 0 & \text{if } x \in -\mathcal{K}^n, \\ u & \text{otherwise}, \end{cases} \qquad \text{where} \qquad u = \begin{bmatrix} \frac{x_1 + \|x_2\|}{2} \\ \left(\frac{x_1 + \|x_2\|}{2}\right) \frac{x_2}{\|x_2\|} \end{bmatrix}.$$

In the same way, x_- can be characterized as follows:

$$x_- = \begin{cases} 0 & \text{if } x \in \mathcal{K}^n, \\ -x & \text{if } x \in -\mathcal{K}^n, \\ w & \text{otherwise}, \end{cases} \qquad \text{where} \qquad w = \begin{bmatrix} -\frac{x_1 - \|x_2\|}{2} \\ \left(\frac{x_1 - \|x_2\|}{2}\right) \frac{x_2}{\|x_2\|} \end{bmatrix}.$$

By direct calculation, it is easy to show that $x = x_+ - x_-$. Moreover, together with the spectral decomposition of x, x_+ and x_- can be described by the following form, respectively:

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

and

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

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

Now, we talk about the expression of $|x|$ associated with $\mathcal{K}^n$. Indeed, we can use the so-called SOC-function to achieve the expression of $|x|$, which can be found in [3, 4, 5]. More specifically, for any $x \in \mathbb{R}^n$, the definition of $|x|$ with respect to SOC is $|x| := x_+ + x_-$. In particular, the form $|x| = x_+ + x_-$ is equivalent to $|x| = \sqrt{x \circ x}$ in the setting of SOC. In light of the above expression of projection

x_+ and x_-, it is easy to get that the expression of the absolute value $|x|$ associated with SOC is the following form

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

Combining the spectral decomposition of x, there is a more explicit expression of $|x|$ as below:

$$|x| = \begin{cases} \dfrac{1}{2}\begin{bmatrix} |x_1 - \|x_2\|| + |x_1 + \|x_2\|| \\ (|x_1 + \|x_2\|| - |x_1 - \|x_2\||)\frac{x_2}{\|x_2\|} \end{bmatrix} & \text{if} \quad x_2 \neq 0, \\ \begin{bmatrix} |x_1| \\ 0 \end{bmatrix} & \text{if} \quad x_2 = 0. \end{cases}$$

In order to employ Levenberg-Marquardt method for solving the SOCAVE (3), we need to adopt a continuously differentiable function. Due to the non-differentiability of $|\alpha|$ for $\alpha \in \mathbb{R}$, we consider a special smoothing function for the absolute value function $|\alpha|$. To this end, we define the function $\phi_p(\cdot,\cdot) : \mathbb{R}^2 \to \mathbb{R}$ as

$$\phi_p(a,b) := \sqrt[p]{|a|^p + |b|^p}, \quad p > 1. \tag{5}$$

Noting that $b \in \mathbb{R}$ and $b \to 0$ yields $\phi_p(a,b) \to |a|$. Therefore, combining the spectral decomposition of x and the function ϕ_p, it is natural to define a vector-valued smoothing function $\Phi : \mathbb{R}^n \to \mathbb{R}^n$ as

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

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

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

which means the function $\Phi(x(\rho))$ is a uniformly smoothing function of $|x|$ associated with SOC. With this function, for tackling the SOCAVE (3), we further define a function $H(x(\rho)) : \mathbb{R}^n \to \mathbb{R}^n$ by

$$H(x(\rho)) = Ax - \Phi(x(\rho)) - b, \qquad \forall \rho \in \mathbb{R},\ x \in \mathbb{R}^n.$$

Then, from a trivial observation

$$\lim_{\rho \to 0} H(x(\rho)) = 0 \quad \Longleftrightarrow \quad Ax - |x| - b = 0,$$

it indicates that x is a solution to the SOCAVE (3) if and only if x is the limit of the solution $x(\rho)$ to the equation $H(x(\rho)) = 0$ when $\rho \to 0$. In practical implementation, we often take $\rho \in \mathbb{R}_{++}$ and set $\rho \downarrow 0$. In addition, it is not difficult to show that the function $H(x(\rho))$ with any $\rho \neq 0$ is continuously differentiable on $\mathbb{R}^n$. By direct calculation, we can write out the explicit formula of the Jacobian matrix for the function H as below:

$$H_x'(x(\rho)) = A - \Phi_x'(x(\rho)) \tag{6}$$

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

$$\Phi_x'(x(\rho)) = \begin{cases} \dfrac{\mathrm{sgn}(x_1)|x_1|^{p-1}}{\left[\sqrt[p]{\rho^p + |x_1|^p}\right]^{p-1}} I & \text{if } x_2 = 0, \\ \begin{bmatrix} b & c\frac{x_2^T}{\|x_2\|} \\ c\frac{x_2}{\|x_2\|} & aI + (b-a)\frac{x_2 x_2^T}{\|x_2\|^2} \end{bmatrix} & \text{if } x_2 \neq 0, \end{cases}$$

with

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

and the function $\mathrm{sgn}(\cdot)$ is defined by $\mathrm{sgn}(\alpha) := \begin{cases} 1 & \text{if } \alpha > 0, \\ 0 & \text{if } \alpha = 0, \\ -1 & \text{if } \alpha < 0. \end{cases}$ It is noteworthy that the Jacobian matrix is derived from Proposition 4 in [5]. Actually, the Jacobian matrix for the function H is one of the important elements in our considered algorithm. For simplicity, we denote $w := x(\rho)$, from which we obtain the system of nonlinear equation

$$H(w) = 0.$$

Thus, x is a solution of the SOCAVE (3) if and only if x is the limit of the solution w to the equation $H(w) = 0$ when $\rho \to 0$.

3. **Levenberg-Marquardt method.** For the SOCAVE (3), Miao et.al.[26] have obtained that the SOCAVE (3) has a unique solution if all singular values of A exceed 1. In view of this, we suppose that all singular values of A exceed 1 in this paper. It follows that the SOCAVE (3) has a unique solution. Now, we consider the Levenberg-Marquardt method with Armijo line search mentioned in [13] for solving the SOCAVE (3) and show its convergence properties. The general iterative scheme is

$$x_{k+1} = x_k + \alpha_k d_k, \quad k = 0, 1, 2, 3, \cdots,$$

where $\alpha_k \in \mathbb{R}_+$ is a step size, and $d_k \in \mathbb{R}^n$ is the search direction. More specifically, α_k and d_k denote the Levenberg-Marquardt direction (4) and the Armijo line search, respectively. For convenience, we denote the merit function Ψ as the following form:

$$\Psi(w) = \frac{1}{2}\|H(w)\|^2.$$

Algorithm 3.1. [Levenberg-Marquardt Algorithm]

Step 0: *Choose an initial point $w_0 = x_0(\rho_0) \in \mathbb{R}^n$ with any $\rho_0 \in \mathbb{R}_{++}$, and parameters $\beta, \varrho \in (0,1)$, $\gamma \in [1,2]$, $\sigma \in (0, \frac{1}{2})$. Set $k := 0$.*

Step 1: *If $\|H(w_k)\| = 0$, then stop. Otherwise go to step 2.*

Step 2: *Set*

$$\mu_k = \|H(w_k)\|^\gamma, \quad d_k = -(J(w_k)^T J(w_k) + \mu_k I)^{-1} J(w_k)^T H(w_k),$$

where $J(w_k)$ denotes the Jacobian matrix $H'(w_k)$ of $H(w_k)$ at w_k given by (6). If d_k satisfies

$$\|H(w_k + d_k))\| \le \varrho \|H(w_k)\|, \tag{7}$$

then set $x_{k+1}(\rho_k) = w_k + d_k$. Then, set $\rho_{k+1} = \frac{\mu_k}{1+\mu_k}\rho_k$ and $w_{k+1} := x_{k+1}(\rho_{k+1})$. Otherwise go to step 3.

Step 3: *Find m_k be the smallest nonnegative integer m such that*

$$\Psi(w_k + \beta^m d_k) \leq \Psi(w_k) + \sigma\beta^m \nabla\Psi(w_k)^T d_k. \tag{8}$$

Set $\alpha_k := \beta^{m_k}$ and $x_{k+1}(\rho_k) = w_k + \alpha_k d_k$. Then, set $\rho_{k+1} = \alpha_k \rho_k$, $w_{k+1} = x_{k+1}(\rho_{k+1})$, and go to step 4.

Step 4: *Set $k := k + 1$, go back to step 1.*

As indicated in [13], the direction d_k in step 2 is unique. Hence, it is clear that Algorithm 3.1 is well defined. The parameter σ guarantees that the line search (8) makes sense, while ϱ could simplify the algorithm when $\Psi(w)$ drops too fast.

Theorem 3.1. *Suppose that all singular values of A exceed 1 and x^* be a solution of the SOCAVE (3). Then, the function $H(w)$ provides a local error bound on the neighborhood $N(x^*, r)$ for the SOCAVE (3), i.e., there exists a constant $c > 0$ such that*

$$\|w - x^*\| \leq c\|H(w)\|, \quad \forall w \in N(x^*, r).$$

Proof. Since x^* is a solution of the SOCAVE (3), we have $H(w^*) := H(x^*(0)) = 0$. Then, it follows that

$$\begin{aligned} H(w) &= H(w) - H(w^*) = H'(\xi)(w - w^*) \\ &= (A - \Phi'_x(\xi))(w - x^*), \end{aligned}$$

where the third equation holds due to the smoothness of Φ with any $\rho > 0$, and $\xi = (1-t)x^* + tw$ with $0 < t < 1$. Since all singular values of A exceed 1, it follows from Theorem 4.1 in [27] that $H'(\xi) = A - \Phi'_x(\xi)$ is nonsingular. Hence, we have

$$w - x^* = (A - \Phi'(\xi))^{-1} H(w),$$

which implies that

$$\|w - x^*\| \leq \|(A - \Phi'_x(\xi))^{-1} H(w)\| \leq \|(A - \Phi'_x(\xi))^{-1}\| \|H(w)\| = c\|H(w)\|,$$

where $c = \|(A - \Phi'_x(\xi))^{-1}\|$. Then, the proof is complete. □

In order to establish the global convergence results for Levenberg-Marquardt Algorithm 3.1, we assume that $\nabla\Psi(w_k) \neq 0$ for all k, which ensures that the algorithm can generate an infinite sequence $\{w_k\}$.

Theorem 3.2. *Suppose that the sequence $\{w_k\}$ is generated by the Levenberg-Marquardt Algorithm 3.1 with Armijo line search. Then, any accumulation point of the sequence $\{w_k\}$ is a stationary point of Ψ. Moreover, if all singular values of A exceed 1, the sequence $\{w_k\}$ converges to the solution of the SOCAVE (3).*

Proof. Since $d_k = -(J(w_k)^T J(w_k) + \mu_k I)^{-1} J(w_k)^T H(w_k)$ and $\nabla\Psi(w_k) \neq 0$ for all k, we have

$$(\nabla\Psi(w_k))^T d_k = -\left(J(w_k)^T H(w_k)\right)^T \left(J(w_k)^T J(w_k) + \mu_k I\right)^{-1} J(w_k)^T H(w_k) < 0,$$

where the inequality holds because $J(w_k)^T J(w_k) + \mu_k I$ is a positive definite matrix for any $\mu_k > 0$. Together with the formulae (7) and (8), the sequence $\{\Psi(w_k)\}$ is monotonically decreasing. By Theorem 3.1, it is easy to get that the sequence $\{w_k\}$ is bounded. Therefore, there are accumulation points in the sequence $\{w_k\}$. Moreover, from the monotone decreasing of $\{\Psi(w_k)\}$, the sequence $\{\mu_k\}$ is also monotonically decreasing, so it has a limit μ^*. To proceed, we discuss two cases for the sequence $\{\mu_k\}$: (i) $\mu^* = 0$ and (ii) $\mu^* \neq 0$.

Case (i): $\mu^* = 0$, i.e., $\mu_k \to 0$. In this case, we have $H(w_k) \to 0$, which illustrates that any accumulation point of the sequence $\{w_k\}$ is a stationary point of Ψ.

Case (ii): $\mu^* \neq 0$, i.e., $\mu^* > 0$. For convenience, without loss of generality, we assume that the sequence $\{w_k\}$ itself is convergent. By $\mu^* \neq 0$, it follows that $\lim_{k\to\infty} \Psi(w_k) \neq 0$ and

$$\begin{aligned}(\nabla\Psi(w_k))^T d_k &= -(J(w_k)^T H(w_k))^T d_k \\ &= -((J(w_k)^T J(w_k) + \mu_k I)d_k)^T d_k \\ &= -d_k^T (J(w_k)^T J(w_k) + \mu_k I)d_k \qquad (9) \\ &\leq -\mu_k \|d_k\|^2 \\ &\leq -\mu^* \|d_k\|^2.\end{aligned}$$

From (9) and the fact that the gradient $\nabla\Psi(w_k)$ is bounded on the convergent sequence $\{w_k\}$, we have

$$\limsup_{k\to\infty} \|d_k\| < \infty.$$

On the other hand, using $\lim_{k\to\infty} \mu_k = \mu^* > 0$ and $\{\mu_k\}$ being monotonically decreasing, it follows that for all large k, there exists a constant $\kappa > 0$ such that

$$\|J(w_k)^T J(w_k) + \mu_k I\| \leq \kappa.$$

This clearly yields

$$\|d_k\| \geq \frac{\|\nabla\Psi(w_k)\|}{\|J(w_k)^T J(w_k) + \mu_k I\|} \geq \frac{\|\nabla\Psi(w_k)\|}{\kappa}.$$

This together with (9) leads to $\limsup_{k\to\infty} \|d_k\| = 0$ or $\limsup_{k\to\infty} |(\nabla\Psi(w_k))^T d_k| > 0$. If $\limsup_{k\to\infty} \|d_k\| = 0$, it implies that $\lim_{k\to\infty} \|\nabla\Psi(w_k)\| = 0$, i.e., the accumulation point of $\{w_k\}$ is a stationary point of Ψ. If $\limsup_{k\to\infty} |(\nabla\Psi(w_k))^T d_k| > 0$, by $\limsup_{k\to\infty} \|d_k\| < \infty$, then we know that the sequence $\{d_k\}$ is uniformly gradient related to $\{w_k\}$. Therefore, by [1, Proposition 1.2.1], we conclude that any accumulation point of $\{w_k\}$ ia a stationary point of the function Ψ.

If all singular values of A exceed 1, it follows that the Jacobian matrix $J(w) = H'(w)$ is nonsingular. By $\nabla\Psi(w) = J(w)H(w)$, it is easy to verify that the stationary point w^* of the function Ψ satisfies $H(w^*) = 0$, which implies that x^* is the solution of the SOCAVE (3). Then, the proof is complete. □

Remark 1. **(a):** Like what is discussed in the literature, the Armijo line search in Algorithm 3.1 could be chosen as Goldstein line search. The corresponding convergence results can be achieved as well.

(b): Based on the strongly semismoothness of the absolute value function associated with SOC, it is easy to get that the Jacobian matrix $H'_x(w)$ is Lipschitz continuous on some neighborhood of the solution x^*. By this, under all singular values of A exceed 1, similar to Theorem 3.1 in [33] or Theorem 2.2 in [10], we get that the sequence $\{w_k\}$ generated by the Levenberg-Marquardt algorithm converges to the solution of SOCAVE (3) quadratically.

4. **Numerical Experiments.** This section is devoted to the numerical implementations. Besides providing some numerical evidence to show the efficiency of Algorithm 3.1, we also do numerical comparison between the Levenberg-Marquardt algorithm and the smoothing Newton algorithm (a promising approach for solving SOCAVE recently studied in [27, 28]). In particular, we adapt the performance profile to describe the influence by perturbing the parameter p.

In our tests, the parameters are set as below:

$$\rho_0 = 0.001, \quad x_0 = \text{rand}(n, 1), \quad \beta = 0.5, \quad \sigma = 0.2, \quad \varrho = 0.5 \quad \text{and} \quad \gamma = 1.$$

We stop the iterations when $\|\nabla\Psi(w_k)\| \leq 10^{-5}$ or the number of iterations exceeds 100. All the experiments are done on a PC with Intel(R) CPU of 2.30GHz and RAM of 4.00GB, and all the program codes are written in Matlab and run in Matlab environment.

To present numerical results, we consider the performance profile which is introduced in [8]. In other words, we regard Algorithm 3.1 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 $\mathcal{P}$. We use the computing time as performance measure for Algorithm 3.1 with different values of p which satisfies the condition $p > 1$ in (5). For each problem q and solver s, let

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

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 a parameter r_M such that $r_{q,s} \leq r_M$ 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

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

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 $r_{q,s}$ is within a factor $\tau \in \mathbb{R}$ of the best possible ratio, see [8].

Additionally, we provide the performance profile in each problem to compare Levenberg-Marquardt algorithm and the smoothing Newton algorithm, which was recently studied in [27, 28] for solving SOCAVE (3) as well. The performance profiles of problem 4.1-4.3 use 300 test problems and problem 4.4 use 40 test problems, they are all with fixed value of $p = 2$ with $n = 300$.

Problem. *Consider the SOCAVE (3) which is generated in the following way: first choose a random matrix $C \in \mathbb{R}^{n\times n}$ from a uniformly distribution on $[-10, 10]$ for every element. We compute the minimal singular value σ_* of C, and let $\sigma := \min\{1, \sigma_*\}$. 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 value of A exceeds 1, which can ensure that the SOCAVE (3) has the unique solution. We choose randomly $b \in \mathbb{R}^n$ on $[0, 1]$ for every element. By Algorithm 3.1 in this paper, the resulting SOCAVE (3) 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 [12].*

Problem. *Consider the SOCAVE (3) which is generated in the following way: choose two random matrices $A \in \mathbb{R}^{n\times n}$ from a uniformly distribution on $[-10, 10]$ for every element. In order to ensure that the SOCAVE (3) is solvable, we update the matrix A by the following: let $[USV] = svd(A)$. If $\min\{S(i, i)\} = 0$ for $i = 0, 1, \cdots, n$, we make $A = U(S + 0.01E)V$, and then $A = \frac{1.01}{\lambda_{\min}(A^TA)}A$. We choose*

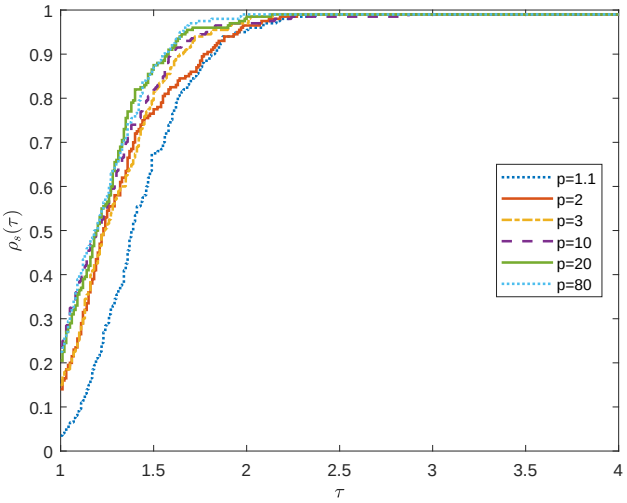

FIGURE 1. Performance profile of computing time of Problem 4.1 with different p.

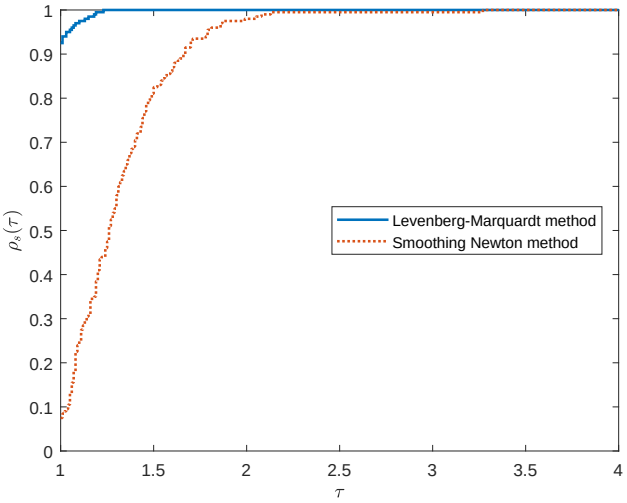

FIGURE 2. Performance profile of computing time of Problem 4.1 with LM and SN methods.

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. *We consider the SOCAVE (3) which is generated by the same way as in Problem 4. However, here the* $\mathcal{K}$ *is Cartesian product of single SOCs, given by* $\mathcal{K} := \mathcal{K}^{n_1} \times \cdots \times \mathcal{K}^{n_r}$*, where* $n_1 = \cdots = n_r = \frac{n}{r}$.

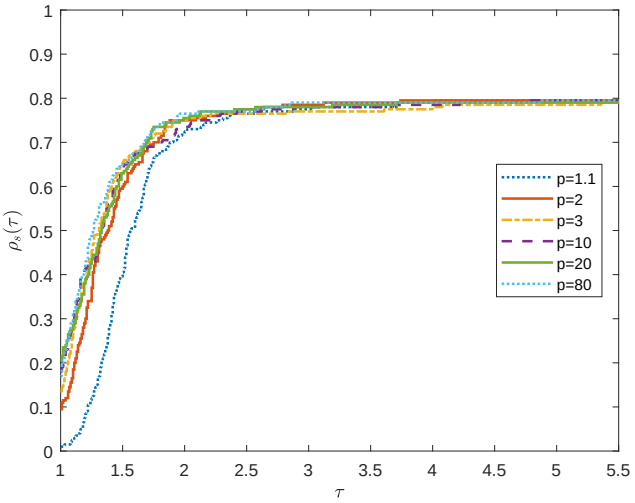

FIGURE 3. Performance profile of computing time of Problem 4.2 with different p.

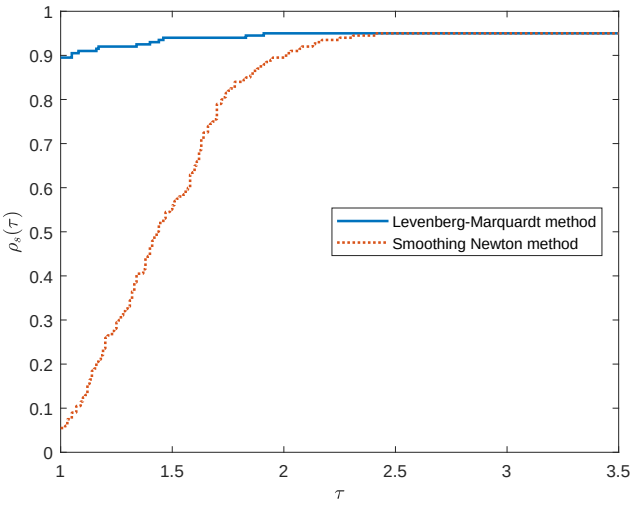

FIGURE 4. Performance profile of computing time of Problem 4.2 with LM and SN methods.

As seen, the above problems 4–4 are all generated randomly. In the below example, we consider the case where the matrix A and the vector b in the SOCAVE (3) are fixed.

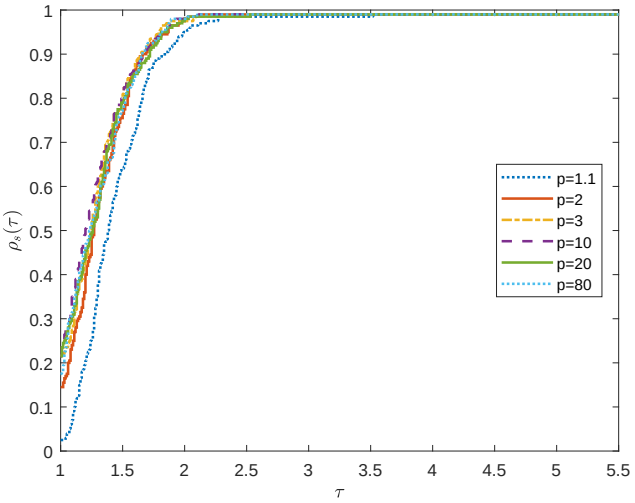

FIGURE 5. Performance profile of computing time of Problem 4.3 with different p.

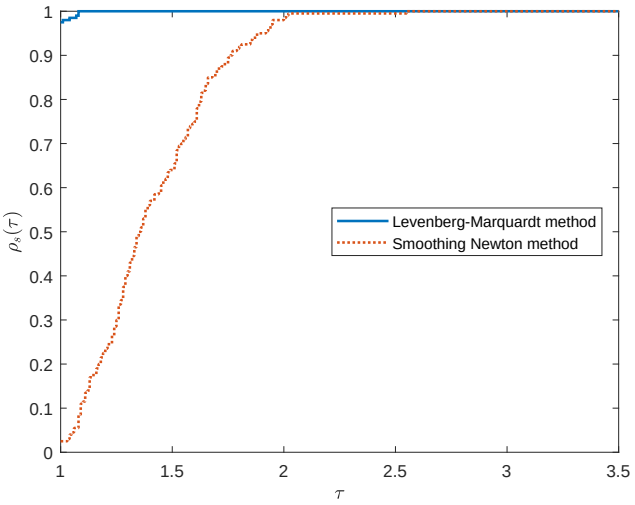

FIGURE 6. Performance profile of computing time of Problem 4.3 with LM and SN methods.

Problem. *Consider the SOCAVE (3) which is generated in the following way: Let*

$$A = \begin{pmatrix} 3 & 2 & 2 & \cdots & 2 \\ 0 & 3 & 2 & \cdots & 2 \\ 0 & 0 & 3 & \cdots & 2 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 2 & 2 & 2 & \cdots & 3 \end{pmatrix}, \quad b = (-2, 2, -2, \ldots, -2, 2 \cdots)^T.$$

The initial point is chosen in the range $[0, 1]$ *entry-wisely. It is obvious that the minimum singular value of* A *exceeds* 1*, so the SOCAVE (3) has the unique solution. In this example, the dimension of* A *is 40.*

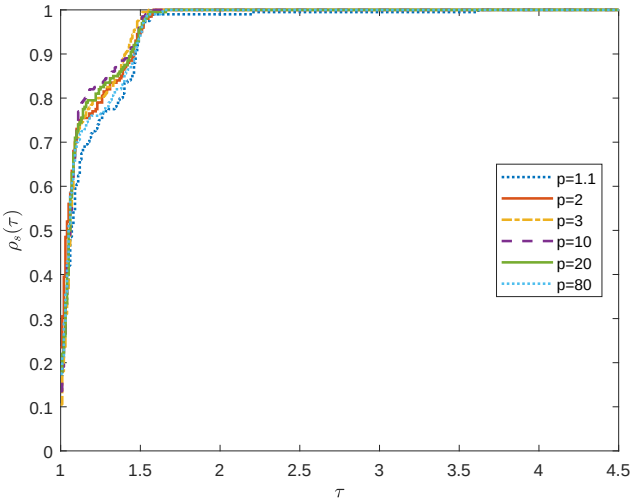

FIGURE 7. Performance profile of computing time of Problem 4.4 with different p.

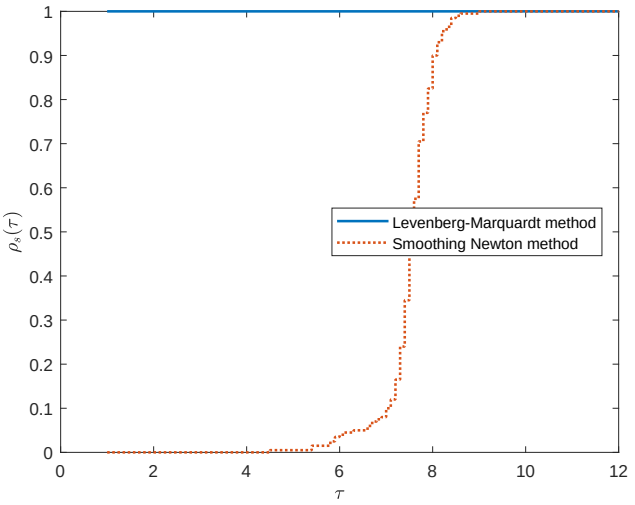

FIGURE 8. Performance profile of computing time of Problem 4.4 with LM and SN methods.

In our experiments, every set of the simulations for every problem is randomly generated ten times, and both of the success rates with Levenberg-Marquardt algorithm and smoothing Newton method are 100%.

We summarize our numerical observations as below:

- From Figures 1,3,5,7, we see that the proposed algorithm is not affected when p is perturbed.
- The small value of p, which is close to 1, seems not a good choice of being employed to work with the proposed algorithm.
- In general, the iterations of Levenberg-Marquardt algorithm are more than those in smoothing Newton method in each problem. Nonetheless, from Figure 1 to Figure 4, we see that the performance in computing time of the Levenberg-Marquardt algorithm is evidently better than the smoothing Newton algorithm.
- All the above results show the efficiency of the Levenberg-Marquardt algorithm. Like the smoothing Newton method, this study suggests that the Levenberg-Marquardt algorithm could be another good choice for solving SOCAVE.

Acknowledgments. We would like to thank the guest editors for giving us an opportunity to contribute this paper to the special issue in memory of Prof. Hang-Chin Lai.

REFERENCES

[1] D. P. Bertsekas, *Nonlinear Programming*, Athena Scientific, Massachusetts, 1995.
[2] L. Caccetta, B. Qu and G.-L. Zhou, A globally and quadratically convergent method for absolute value equations, *Computational Optimization and Applications*, **48** (2011), 45–58.
[3] J. S. Chen, The convex and monotone functions associated with second-order cone, *Optimization*, **55** (2006), 363–385.
[4] J. S. Chen, *SOC Functions and Their Applications*, Springer Optimization and Its Applications 143, Springer, Singapore, 2019.
[5] J. S. Chen, X. Chen and P. Tseng, Analysis of nonsmooth vector-valued functions associated with second-order cones, *Mathematical Programming*, **101** (2004), 95–117.
[6] J. S. Chen and S. H. Pan, A survey on SOC complementarity functions and solution methods for SOCPs and SOCCPs, *Pacific Journal of Optimization*, **8** (2012), 33–74.
[7] J. S. Chen and P. Tseng, An unconstrained smooth minimization reformulation of second-order cone complementarity problem, *Mathematical Programming*, **104** (2005), 293–327.
[8] E. D. Dolan and J. J. More, Benchmarking optimization software with performance profiles, *Mathematical Programming*, **91** (2002), 201–213.
[9] U. Faraut and A. Koranyi, *Anlysis on Symmetric Cones*, Oxford Mathematical Monographs, Oxford University Press, New York, 1994.
[10] J.-Y. Fan and Y.-X. Yuan, On the quadratic convergence of the Levenberg-Marquardt method without nonsingularity assumption, *Computing*, **74** (2005), 23–39.
[11] S.-L. Hu and Z.-H. Huang, A note on absolute value equations, *Optimization Letters*, **4** (2010), 417–424.
[12] S.-L. Hu, Z.-H. Huang and Q. Zhang, A generalized Newton method for absolute value equations associated with second order cones, *Journal of Computational and Applied Mathematics*, **235** (2011), 1490–1501.
[13] J. Iqbal, A. Iqbal and M. Arif, Levenberg-Marquardt method for solving systems of absolute value equations, *Journal of Computational and Applied Mathematics*, **282** (2015), 134–138.
[14] X.-Q. Jiang, A smoothing newton method for solving absolute value equations, *Advanced Materials Research*, **765-767** (2013), 703–708.
[15] X.-Q. Jiang and Y. Zhang, A smoothing-type algorithm for absolute value equations, *Journal of Industrial and Management Optimization*, **9** (2013), 789–798.
[16] S. Ketabchi and H. Moosaei, Minimum norm solution to the absolute value equation in the convex case, *Journal of Optimization Theory and Applications*, **154** (2012), 1080–1087.
[17] K. Levenberg, A method for the solution of certain nonlinear problems in least squares, *Quarterly of Applied Mathematics*, **2** (1944), 164–168.

[18] O. L. Mangasarian, Absolute value programming, *Computational Optimization and Applications*, **36** (2007), 43–53.
[19] O. L. Mangasarian, Absolute value equation solution via concave minimization, *Optimization Letters*, **1** (2007), 3–5.
[20] O. L. Mangasarian, A generalized Newton method for absolute value equations, *Optimization Letters*, **3** (2009), 101–108.
[21] O. L. Mangasarian, Absolute value equation solution via dual complementarity, *Optimization Letters*, **7** (2013), 625–630.
[22] O. L. Mangasarian, Linear complementarity as absolute value equation solution, *Optimization Letters*, **8** (2014), 1529–1534.
[23] O. L. Mangasarian, Absolute value equation solution via linear programming, *Journal of Optimization Theory and Applications*, **161** (2014), 870–876.
[24] O. L. Mangasarian and R. R. Meyer, Absolute value equation, *Linear Algebra and Its Applications*, **419** (2006), 359–367.
[25] D. W. Marquardt, An algorithm for least squares estimation of nonlinear inequalities, *Journal of the Society for Industrial and Applied Mathematics*, **11** (1963), 431–441.
[26] X.-H. Miao, W.-M. Hsu, C. T. Nguyen and J.-S. Chen, The solvabilities of three optimization problems associated with second-order cone, *Journal of Nonlinear and Convex Analysis*, **22** (2021), 937–967.
[27] X.-H. Miao, J.-T. Yang, B. Saheya and J.-S. Chen, A smoothing Newton method for absolute value equation associated with second-order cone, *Applied Numerical Mathematics*, **120** (2017), 82–96.
[28] C. T. Nguyen, B. Saheya, Y.-L. Chang and J.-S. Chen, Unified smoothing functions for absolute value equation associated with second-order cone, *Applied Numerical Mathematics*, **135** (2019), 206–227.
[29] O. A. Prokopyev, On equivalent reformulations for absolute value equations, *Computational Optimization and Applications*, **44** (2009), 363–372.
[30] J. Rohn, A theorem of the alternative for the equation $Ax + B|x| = b$, *Linear and Multilinear Algebra*, **52** (2004), 421–426.
[31] J. Rohn, An algorithm for solving the absolute value equation, *Eletronic Journal of Linear Algebra*, **18** (2009), 589–599.
[32] S. Yamanaka and M. Fukushima, A brancd and bound method for the absolute value programs, *Optimization*, **63** (2014), 305–319.
[33] N. Yamashita and M. Fukushima, On the rate of convergence of the Levenberg-Marquardt method, in *Topics in Nonlinear Analysis* (eds. by G. Alefeld and X. Chen), Springer-Verlag, (2001), 239–249.
[34] X. Zhu and G.-H. Lin, Improved convergence results for a modified Levenberg-Marquardt method for nonlinear equations and applications in MPCC, *Optimization Methods and Software*, **31** (2016), 791–804.
[35] C. Zhang and Q.-J. Wei, Global and finite convergence of a generalized Newton method for absolute value equations, *Journal of Optimization Theory and Applications*, **143** (2009), 391–403.

Received February 2020; revised October 2020; Final revision January 2021; Early access November 2021.

E-mail address: xinhemiao@tju.edu.cn
E-mail address: yaokai@tju.edu.cn
E-mail address: yangcy@math.ntnu.edu.tw
E-mail address: jschen@math.ntnu.edu.tw