symmetry

MDPI

Article

A Modified Spectral Conjugate Gradient Method for Absolute Value Equations Associated with Second-Order Cones

Leifu Gao [1,2,*], Zheng Liu [1,2], Jingfei Zou [2] and Zengwei Wang [2]

[1] School of Business Administration, Liaoning Technical University, Huludao 125105, China
[2] Institute for Optimization and Decision Analytics, Liaoning Technical University, Fuxin 123000, China
* Correspondence: lfgao@lntu.edu.cn

Abstract: In this paper, we propose a modified spectral conjugate gradient (MSCG) method for solving absolute value equations associated with second-order cones (SOCAVEs). Some properties of the SOCAVEs are analyzed, and the global convergence of the MSCG method is discussed in depth. Numerical experiments are given to illustrate the effectiveness and competitiveness of our algorithm.

Keywords: absolute value equation; second-order cone; spectral conjugate gradient method; convergence analysis

check for updates

Citation: Gao, L.; Liu, Z.; Zou, J.; Wang, Z. A Modified Spectral Conjugate Gradient Method for Absolute Value Equations Associated with Second-Order Cones. *Symmetry* **2024**, *16*, 654. https://doi.org/10.3390/sym16060654

Academic Editor: Calogero Vetro

Received: 7 April 2024
Revised: 9 May 2024
Accepted: 13 May 2024
Published: 25 May 2024

Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

In this paper, we focus on investigating the following absolute value equations associated with second-order cones (SOCAVEs)

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

where $A \in \mathbb{R}^{n\times n}$, $b \in \mathbb{R}^n$, and $|x| \in \mathcal{K}^n$. The second-order cone (SOC) is characterized by the following mathematical representation

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

If $n = 1$, let $\mathcal{K}^n$ be the set of non-negative reals $\mathbb{R}_+$. In addition, the general SOC $\mathcal{K}$ can be considered as the Cartesian product of individual SOCs, that is,

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

where $n_1 + \cdots + n_r = n$ and $n_i (i = 1, \cdots, r)$ is a non-negative integer. Without loss of generality, we will concentrate on the scenario where $r = 1$, as all the analysis can be extended to the case of $r > 1$ based on the characteristics of the Cartesian product. For every $x = (x_1, x_2) \in \mathbb{R} \times \mathbb{R}^{n-1}$ and $y = (y_1, y_2) \in \mathbb{R} \times \mathbb{R}^{n-1}$, their Jordan product $\circ$ is defined as [1–3]

$$x \circ y = (\langle x, y\rangle, y_1 x_2 + x_1 y_2) \in \mathbb{R} \times \mathbb{R}^{n-1}.$$

Based on this definition, the absolute value vector $|x|$ in SOC $\mathcal{K}^n$ can be calculated by

$$|x| = \sqrt{x \circ x}. \tag{2}$$

In the context of solving SOCAVEs (1), the definition of $|x|$ is specified in (2). SOCAVEs (1) can be seen as a particular form of the generalized absolute value equations associated with SOCs (SOCGAVEs)

$$Cx + D|x| - c = 0 \tag{3}$$

with $C, D \in \mathbb{R}^{m\times n}$, $c \in \mathbb{R}^m$, and $|x| \in \mathcal{K}^n$. Actually, SOCGAVEs (3) was first introduced in [4] and then further explored in subsequent research conducted by [5–8], as well as the references therein. Furthermore, SOCAVEs (1) can be viewed as a generalization of absolute value equations (AVEs)

$$Ax - |x| = b, \tag{4}$$

where $A \in \mathbb{R}^{n\times n}$ and $b \in \mathbb{R}^n$. Meanwhile, SOCGAVEs (3) can be seen as an extension of generalized AVEs (GAVEs)

$$Cx + D|x| = c, \tag{5}$$

where $C, D \in \mathbb{R}^{m\times n}$and $c \in \mathbb{R}^m$. In AVEs (4) and GAVEs (5), $|x| = (|x_1|, |x_2|, \cdots, |x_n|)^T \in \mathbb{R}^n$. It is worth noting that GAVEs (5) was initially introduced in [9] and subsequently explored in [10–12], along with the references included therein. Undoubtedly, AVEs (4) can be regarded as a particular case of GAVEs (5).

In recent years, AVEs (4) and GAVEs (5) have been extensively studied due to their significance in a variety of mathematical programming problems, including the linear complementarity problem (LCP), bimatrix game and others; for further details, refer to [10,11,13]. As a result, numerous theoretical findings and numerical algorithms have been developed for both AVEs (4) and GAVEs (5). In term of theory, we elaborate on two aspects which are the equivalent reformulations of AVEs (4) and its solvabilities. As for the equivalent reformulations, ref. [13] proved that AVEs (4) could be reformulated as the LCP when 1 is not a singular value of A, which can subsume many optimization problems such as linear programs, quadratic programs, and bimatrix games. Ref. [11] further improved the above equivalence which indicates that AVEs (4) can be equivalently recast as the LCP without any assumption on A and B, and the authors also provided a relationship with mixed integer programming. Ref. [14] gave the equivalent relation between the horizontal linear complementarity problems (HLCP) and AVEs (4). When considering the existence of a solution, the sufficient conditions of a unique solution, many solutions, and no solution were presented by [13]. Ref. [15] presented two necessary and sufficient conditions for the unique solution of AVEs (4). That is, matrix $A - I + 2D$ or $A + I - 2D$ are nonsingular for any diagonal matrix $D = diag(d_i)$ with $0 \leq d_i \leq 1$. Based on the equivalent reformulation between AVEs (4) and the HLCP, ref. [14] also gave the necessary and sufficient condition for the unique solvability of AVEs (4), i.e., $\{A - I, A + I\}$ has the column $\mathcal{W}$-property. Under the condition that AVEs (4) has a unique solution, ref. [16] conducted a study on the error bound and condition number of AVEs (4), which play crucial roles for the convergence analysis of AVEs (4). Ref. [12] analyzed topological properties of a solution set for AVEs (4), including convexity, boundedness, or whether it consists of many finite solutions.

When considering numerical algorithms, ref. [10] proposed a generalized Newton method for solving AVEs (4). Based on the equivalent reformulation between AVEs (4) and the two-by-two block nonlinear equations, ref. [17] developed the SOR-like iterative method and [18] analyzed the selection of optimal parameter for this algorithm. By reformulating AVEs (4) into a new nonlinear system, ref. [19] put forward an alternative SOR-like method to solve AVEs (4). Subsequently, many researchers, such as [20–22], presented many iterative methods by applying the idea of [17]. Furthermore, AVEs (4) can be seen as a class of special nonlinear equations; based on this, ref. [23] proposed a modified multivariate spectral gradient algorithm for AVEs (4) and [24] improved a new three-term spectral subgradient method for solving AVEs (4).

Our interest in SOCAVEs (1) and SOCGAVEs (3) is motivated by their status as extensions of conventional forms, as well as their equivalence to LCPs associated with SOC (SOCLCPs). These problems have wide-ranging applications in engineering, control systems, and finance, as evidenced by studies such as [4,6,7]. Notably, recent research efforts have yielded significant developments in both numerical methodologies and theoretical findings for addressing SOCAVEs (1) and SOCGAVEs (3). Ref. [4] have shown that SOCAVEs (1) is equivalent with the second-order cone linear complementarity problem (SOCLCP). Furthermore, ref. [6] have proved that SOCAVEs (1) can be converted into

the standard SOCLCP. When considering the existence of a solution, ref. [25] presented some sufficient conditions for the unique solution of SOCAVEs (1). Furthermore, by using P-property and a globally uniquely solvable (GUS) property in the SOCLCP, ref. [26] showed some sufficient and necessary conditions for the unique solution of SOCAVEs (1). When considering the numerical algorithms for solving SOCAVEs (1), ref. [4] proposed a generalized Newton method and demonstrated that the proposed approach exhibits global linear convergence and local quadratic convergence under appropriate assumptions. Considering the nonlinear and non-smoothing term of $|x| \in \mathcal{K}^n$, by applying different smoothing techniques, the smoothing-type algorithms have been proposed by [6–8] for SOCAVEs (1). The above smoothing-type algorithms usually apply the monotone line search technique. In order to improve the computing performance of the smoothing-type algorithms, ref. [27] introduced a non-monotone smoothing Newton algorithm for solving SOCAVEs (1). Based on a splitting of the two-by-two block coefficient matrix, ref. [28] proposed a modified SOR-like method to solve SOCAVEs (1).

SOCAVEs (1) can be interpreted as a special case of the following system of nonlinear equations

$$F(x) = 0 \quad \text{with} \quad F(x) := Ax - |x| - b, \tag{6}$$

where $F : \mathbb{R}^n \to \mathbb{R}^n$ is continuous and monotone. $F(x)$ is called a monotone equation such that it satisfies

$$\langle F(x) - F(y), x - y \rangle \geq 0, \forall x, y \in \mathbb{R}^n. \tag{7}$$

For the general nonlinear equations, the spectral conjugate gradient methods have garnered significant attention since they do not require the first order information at each iteration. Moreover, the spectral conjugate gradient methods have been successfully applied to find local minimizers of larger-scale problems. Using these motivations, in this paper, we develop a MSCG method for solving SOCAVEs (1).

In particular, the significant contributions of this paper can be summarized as follows.

(i) A modified spectral conjugate gradient method is proposed for solving SOCAVEs (1).
(ii) The properties of the objective function of SOCAVEs (1) under suitable conditions are established.
(iii) In comparison to the spectral gradient algorithm, the proposed method is appropriate for resolving SOCAVEs (1) due to its low storage demands and exclusive reliance on the value of objective function.
(iv) Numerical examples are given to demonstrate the effectiveness of the proposed method.

The remainder of this paper is structured as follows. Section 2 provides an overview of the preliminaries, propositions, and lemmas that are essential for understanding and analyzing the subsequent content. In Section 3, we propose a modified spectral conjugate gradient method for solving SOCAVEs (1). Furthermore, the global convergence of the proposed method is presented in Section 4. In Section 5, numerical examples are given to illustrate the efficiency of the proposed algorithm. Concluding remarks are made in Section 6.

2. Preliminaries

In this section, we gather fundamental results essential for our subsequent analysis.

We begin by reviewing key concepts and background information concerning SOCs, as detailed in [1–3,29,30].

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

$$x = \lambda_1 \mu_1 + \lambda_2 \mu_2, \tag{8}$$

where λ_1, λ_2 and μ_1, μ_2 are the spectral values and the corresponding spectral vectors of x defined as

$$\lambda_i = x_1 + (-1)^i \|x_2\|$$

and

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

for $i = 1, 2$, with $\omega \in \mathbb{R}^{n-1}$ and satisfying $\|\omega\| = 1$. Indeed, the decomposition (8) is guaranteed to be unique if $x_2 \neq 0$.

To derive a more explicit formula for $|x| \in \mathcal{K}^n$, we need to give some results about the functions associated with SOC.

Definition 1. *For any function $\hat{f} : \mathbb{R} \to \mathbb{R}$, defining a function on $\mathbb{R}^n$ associated with $\mathcal{K}^n$ $(n \geq 1)$ as*

$$f(x) = \hat{f}(\lambda_1)\mu_1 + \hat{f}(\lambda_2)\mu_2, \forall x = (x_1, x_2) \in \mathbb{R} \times \mathbb{R}^{n-1}, \tag{9}$$

where λ_1 and λ_2 are the spectral values and μ_1 and μ_2 are the associated spectral vectors of x.

If $n = 1$, according to (8), we have $\lambda_1 = \lambda_2 = x$ and $\mu_1 = \mu_2 = \frac{1}{2}$. Therefore, for any $x \in \mathbb{R}$, the spectral decomposition of x is $x = \frac{1}{2}x + \frac{1}{2}x$. By the definition of the SOC function (9), it holds that $f(x) = \frac{1}{2}\hat{f}(x) + \frac{1}{2}\hat{f}(x) = \hat{f}(x)$ for all $x \in \mathbb{R}$.

Next, we present some special examples about the function associated with SOC. For any $x = (x_1, x_2) \in \mathbb{R} \times \mathbb{R}^{n-1}$, according to the spectral decomposition (8), we have

$$x^2 = \lambda_1^2\mu_1 + \lambda_2^2\mu_2 \tag{10}$$

and

$$\sqrt{x} = \sqrt{\lambda_1}\mu_1 + \sqrt{\lambda_2}\mu_2. \tag{11}$$

When $n = 1$, we have $\lambda_1 = \lambda_2 = x$ and $\mu_1 = \mu_2 = \frac{1}{2}$. According to (11), it holds that $\sqrt{x} = \frac{1}{2}\sqrt{x} + \frac{1}{2}\sqrt{x}$.

By using (10) and (11), we have

$$\begin{aligned} \sqrt{x^2} &= \sqrt{\lambda_1^2\mu_1 + \lambda_2^2\mu_2} \\ &= \sqrt{\lambda_1^2}\mu_1 + \sqrt{\lambda_2^2}\mu_2 \\ &= |\lambda_1|\mu_1 + |\lambda_2|\mu_2. \end{aligned}$$

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

$$|x| = |\lambda_1|\mu_1 + |\lambda_2|\mu_2. \tag{12}$$

According to (8) and (12), we have

$$|x| = \begin{cases} \frac{1}{2}\left(|x_1 - \|x_2\|| + |x_1 + \|x_2\||, (|x_1 + \|x_2\|| - |x_1 - \|x_2\||)\frac{x_2}{\|x_2\|}\right), & \text{if } x_2 \neq 0, \\ (|x_1|, 0), & \text{if } x_2 = 0. \end{cases} \tag{13}$$

Finally, we discuss the Lipschitz continuous property and monotonicity of the mapping F (6). First, we give the following definitions and the auxiliary lemma.

Definition 2. *The mapping F (6) is Lipschitz continuous; a constant $L > 0$ exists and satisfies*

$$\|F(x) - F(y)\| \leq L\|x - y\|, \forall x, y \in \mathbb{R}^n. \tag{14}$$

Definition 3. *The mapping F (6) is strongly monotone; a constant $m > 0$ exists such that*

$$\langle F(x) - F(y), x - y\rangle \geq m\|x - y\|^2, \forall x, y \in \mathbb{R}^n. \tag{15}$$

Lemma 1 ([31]). *For any vectors $x = (x_1, x_2)$ and $y = (y_1, y_2) \in \mathbb{R} \times \mathbb{R}^{n-1}$, then*

$$\| |x| - |y| \| \leq \|x - y\|. \tag{16}$$

Proposition 1. *The mapping F (6) is Lipschitz continuous.*

Proof. For $\forall x, y \in \mathbb{R}^n$, we have

$$\begin{aligned} \|F(x) - F(y)\| &= \|Ax - |x| - Ay + |y|\| \\ &= \|A(x - y) - (|x| - |y|)\| \\ &\leq \|A(x - y)\| + \| |x| - |y| \| \\ &\leq \|A\| \|x - y\| + \|x - y\| \\ &= (\|A\| + 1)\|x - y\|. \end{aligned}$$

Therefore, the mapping F (6) is Lipschitz continuous with $L = \|A\| + 1$. □

Proposition 2. *Assume $A - I$ is positive semi-definite matrix, then the mapping F (6) is monotone.*

Proof. For $\forall x, y \in \mathbb{R}^n$, we have

$$\begin{aligned} &\langle F(x) - F(y), x - y \rangle \\ &= \langle A(x - y) - (|x| - |y|), x - y \rangle \\ &= (x - y)^T A(x - y) - (x - y)^T (|x| - |y|). \end{aligned}$$

Using the Cauchy–Schwarz inequality, we have

$$\begin{aligned} (x - y)^T (|x| - |y|) &\leq ||x - y|| \cdot \| |x| - |y| \| \\ &\leq \|x - y\|^2, \end{aligned} \tag{17}$$

in which Lemma 1 is used. Then,

$$\begin{aligned} \langle F(x) - F(y), x - y \rangle &= (x - y)^T A(x - y) - (x - y)^T (|x| - |y|) \\ &\geq (x - y)^T A(x - y) - \|x - y\|^2 \\ &= (x - y)^T (A - I)(x - y). \end{aligned} \tag{18}$$

Obviously, when $A - I$ positive semi-definite matrix, the mapping F (6) is monotone. □

Proposition 3. *If A is a symmetric matrix and $A - I$ is positive definite, then the mapping F (6) is strongly monotone.*

Proof. Owing to the last inequality of (18), that is

$$\langle F(x) - F(y), x - y \rangle \geq (x - y)^T (A - I)(x - y),$$

then we have

$$\begin{aligned}\langle F(x)-F(y),x-y\rangle &\geq \frac{1}{2}\Big[(x-y)^T(A-I)(x-y)+(x-y)^T(A-I)^T(x-y)\Big]\\ &=\frac{1}{2}(x-y)^T(A+A^T)(x-y)-(x-y)^T(x-y)\\ &\geq \frac{1}{2}\lambda_{\min}(A+A^T)\|x-y\|^2-\|x-y\|^2\\ &=(\frac{1}{2}\lambda_{\min}(A+A^T)-1)\|x-y\|^2\\ &=\big(\lambda_{\min}(A)-1\big)\|x-y\|^2,\end{aligned}$$

in which $\lambda_{\min}(A)$ denotes the smallest eigenvalue of A; thus, the last equation holds by the symmetry property of matrix A. Because $A-I$ is definitely positive, then $\lambda_{\min}(A)-1>0$. Therefore, the mapping F (6) is strongly monotone. □

3. The Algorithm

In this section, we develop a modified spectral conjugate gradient method for solving SOCAVEs (1).

The spectral conjugate gradient method is a class of important optimization methods for solving nonlinear equations. Let us simply describe the algorithm framework of the spectral conjugate gradient method for nonlinear Equation (6). Starting from an initial guess $x_0 \in \mathbb{R}^n$, the spectral conjugate gradient method usually generates a sequence $\{x_k\}$ satisfying

$$x_{k+1}=x_k+\alpha_k d_k,\ k\geq 0, \tag{19}$$

where $\alpha_k>0$ is the step size which is obtained by a suitable line search, and the search direction d_k is defined as

$$d_k=\begin{cases}-F(x_k), & \text{if}\quad k=0,\\ -\theta_k F(x_k)+\beta_k d_{k-1}, & \text{if}\quad k\geq 1,\end{cases} \tag{20}$$

where θ_k and β_k are the spectral parameter and the conjugate parameter, respectively. Therefore, the performance property of spectral conjugate gradient methods depends on the choice of the parameters θ_k and β_k. For the sake of convenience, we use F_k to replace $F(x_k)$. Inspired by [32], we choose the following parameters with

$$\theta_k=-\frac{F_{k-1}^T d_{k-1}}{\|F_{k-1}\|^2}+\beta_k\frac{F_k^T d_{k-1}}{\|F_k\|^2} \tag{21}$$

and

$$\beta_k=\frac{\sigma F_k^T y_{k-1}}{\|F_k\|\|d_{k-1}\|}, \tag{22}$$

where $y_{k-1}=F_k-F_{k-1}$.

In what follows, we will demonstrate that the search direction d_k generated by (20)–(22) is a sufficient descent direction, which satisfies

$$F_k^\top d_k=-\|F_k\|^2. \tag{23}$$

In general, the formula of the sufficient descent condition is

$$F_k^\top d_k\leq -\gamma\|F_k\|^2,$$

where $\gamma>0$ is a positive constant. In contrast to the above condition, an attractive property of the sufficient descent direction (23) is its independence from any line search. Moreover, if the line search is exact, then $F_k^\top d_{k-1}=0$. In this case, we have

$$\theta_k = -\frac{F_{k-1}^T d_{k-1}}{\|F_{k-1}\|^2} = 1,$$

then the MSCG method can reduce to the standard conjugate gradient method.

Lemma 2. *The search direction d_k defined by (20)–(22) is the sufficient descent direction, i.e., the condition (23) holds for $k \geq 0$.*

Proof. Pre-multiplying (20) by $F_k^\top$, we have

$$\begin{aligned} F_k^\top d_k &= -\theta_k \|F_k\|^2 + \beta_k F_k^\top d_{k-1} \\ &= \|F_k\|^2 \cdot \frac{F_{k-1}^\top d_{k-1}}{\|F_{k-1}\|^2} \cdot \left(-\frac{\|F_{k-1}\|^2}{F_{k-1}^\top d_{k-1}} \theta_k + \beta_k \frac{\|F_{k-1}\|^2}{F_{k-1}^\top d_{k-1}} \cdot \frac{F_k^\top d_{k-1}}{\|F_k\|^2} \right). \end{aligned}$$

Let $\delta_k := -\frac{\|F_{k-1}\|^2}{F_{k-1}^\top d_{k-1}} \theta_k + \beta_k \frac{\|F_{k-1}\|^2}{F_{k-1}^\top d_{k-1}} \cdot \frac{F_k^\top d_{k-1}}{\|F_k\|^2}$. Applying (21), it is easy to deduce that $\delta_k = 1$.

Therefore, we have

$$\frac{F_k^\top d_k}{\|F_k\|^2} = \frac{F_{k-1}^\top d_{k-1}}{\|F_{k-1}\|^2} = \cdots = \frac{F_0^\top d_0}{\|F_0\|^2} = -1,$$

and the condition (23) is satisfied. □

This means that if we can choose the search direction d_k to satisfy $\delta_k \equiv 1$, then d_k always satisfies the sufficient descent property. Moreover, this remarkable property is independent of any line search.

The above analysis allows us to obtain a reasonable parameter θ_k. To obtain a global convergent algorithm for solving nonlinear monotone Equation (6), the search direction generated by the algorithm is required to be bounded. At this point, we describe a more specific algorithm as follows.

The following lemma shows that Algorithm 1 is well defined.

Algorithm 1 Modified Spectral Conjugate Gradient Method (MSCG).

Step 0. Choose an initial guess $x_0 \in \mathbb{R}^n$, $\xi, tol > 0$, $0 < \delta < 1$, $\sigma > 0$ and set $k = 0$.
Step 1. If $\|F_k\| \leq tol$, terminate. Else go to **Step 2**.
Step 2. Compute d_k using (19)–(22).
Step 3. Set the test point μ_k

$$\mu_k = x_k + \alpha_k d_k, \tag{24}$$

where $\alpha_k = \delta^i$, for $i = 0, 1, \cdots$, satisfying

$$-F(x_k + \delta^i d_k)^\top d_k \geq \xi \delta^i \|d_k\|^2. \tag{25}$$

Step 4. Update next iterative point by the hyperplane projection method,

$$x_{k+1} = x_k - \zeta_k F(\mu_k), \tag{26}$$

where

$$\zeta_k = \frac{F(\mu_k)^\top (x_k - \mu_k)}{\|F(\mu_k)\|^2}. \tag{27}$$

Step 5. Set $k := k + 1$, go to **Step 1**.

Lemma 3. *The Algorithm 1 is well defined, i.e., for all k, the step size $\alpha_k = \delta^i$ exists such that the line search condition (25) holds.*

Proof. We prove the conclusion by contradiction. Assume there exists a constant $k_0 \geq 0$ such that (25) does not satisfy for any non-negative integer i, i.e.,

$$-F(x_{k_0} + \delta^i d_{k_0})^\top d_{k_0} < \xi \delta^i \|d_{k_0}\|^2.$$

From Proposition 1 and allowing $i \to \infty$, we have

$$-F(x_{k_0})^\top d_{k_0} \leq 0. \tag{28}$$

Using (23), we can derive that

$$-F(x_{k_0})^\top d_{k_0} = \|F(x_{k_0})\|^2 > 0,$$

which contradicts (28). Therefore, the Algorithm 1 is well defined. □

Remark 1. *By the monotonicity of F(6), we have*

$$\langle F(\mu_k), \bar{x} - \mu_k \rangle = 0,$$

for all $\bar{x}$ such that $F(\bar{x}) = 0$. By applying a line search procedure along the descent direction d_k, we obtain a point $\mu_k = x_k + \alpha_k d_k$ such that

$$\langle F(\mu_k), x_k - \mu_k \rangle > 0.$$

Thus, the hyperplane

$$\mathcal{H}_k = \{x \in R^n | \langle F(\mu_k), x_k - \mu_k \rangle = 0\}$$

strictly separates the current iterate x_k from zeros of Equation (6). Once the separating hyperplane is obtained, the next iteration point x_{k+1} is computed by projecting x_k onto the hyperplane.

4. Convergence Analysis

In this section, we establish the global convergence of Algorithm 1. To this end, we give the following auxiliary lemmas to complete the proof of convergence.

Lemma 4 ([33]). *Suppose that the sequence $\{\mu_k\}$ and $\{x_{k+1}\}$ are generated by (19) and (26) in Algorithm 1. If the mapping F is monotone and satisfying*

$$F(\mu_k)^\top (x_k - \mu_k) > 0, \tag{29}$$

then for any $\bar{x} \in \mathbb{R}^n$ such that $F(\bar{x}) = 0$, it holds that

$$\|x_{k+1} - \bar{x}\|^2 \leq \|x_k - \bar{x}\|^2 - \|x_{k+1} - x_k\|^2. \tag{30}$$

Lemma 5 ([34,35]). *Let $\{\tau_k\}$ be a sequence of real numbers and $\{t_k\}$ be a sequence of real non-negative numbers. Let's assume that the sequence $\{\tau_k\}$ is bounded and for any $k \geq 0$, it satisfies that*

$$\tau_{k+1} \leq \tau_k - t_k,$$

then the following statements hold:

(i) For $k \geq 0$, τ_k is bounded;

(ii) The sequence $\{t_k\}$ is summable, namely $\sum_{k=0}^{\infty} t_k < \infty$;

(iii) The sequence $\{\tau_k\}$ is monotonically decreasing and convergent.

Lemma 6. *Based on Propositions 1 and 2, if* $\{\mu_k\}$ *and* $\{x_k\}$ *are sequences generated by (19) and (26) in Algorithm 1, then the following statements hold:*

(a) The sequence $\{x_k\}$ *is bounded;*
(b) The sequence $\{F_k\}$ *is bounded;*
(c) The step size α_k *and the search direction* d_k *satisfy*

$$\lim_{k\to\infty} \alpha_k \|d_k\| = 0. \tag{31}$$

Proof. Assuming that $\bar{x}$ is a solution of SOCAVEs (1).

(a) From the line search condition (25), we can easily deduce that the inequity (29) holds. Based on Lemmas 4 and 5, we obtain the following results:

- The sequence $\{\|x_{k+1} - \bar{x}\|\}$ is monotonically decreasing and convergent;
- The sequence $\{\|x_{k+1} - x_k\|\}$ is summable, i.e.,

$$\sum_{k=0}^{\infty} \|x_{k+1} - x_k\|^2 < \infty. \tag{32}$$

Based on the aforementioned results, we have

$$\|x_k - \bar{x}\| \leq \|x_0 - \bar{x}\| \tag{33}$$

and the sequence $\{x_k\}$ is bounded, i.e., a constant $c > 0$ exists such that

$$\|x_k\| \leq c, \forall k \geq 0. \tag{34}$$

In addition, based on the inequality (32), we have

$$\lim_{k\to\infty} \|x_{k+1} - x_k\| = 0. \tag{35}$$

(b) By the Lipschitz continuity of F and (33), it follows that

$$\|F_k\| = \|F_k - F(\bar{x})\| \leq L\|x_k - \bar{x}\| \leq L\|x_0 - \bar{x}\|,$$

then, by setting $\varepsilon = \|x_0 - \bar{x}\|$, we have

$$\|F_k\| \leq L\varepsilon. \tag{36}$$

Therefore, the sequence $\{F_k\}$ is bounded.

(c) Based on Proposition 2, we obtain that

$$F(\mu_k)^T(x_k - \mu_k) \leq F_k^T(x_k - \mu_k).$$

From (19) and (25) and the inequality above,

$$\xi\|\mu_k - x_k\| = \xi\frac{\|\mu_k - x_k\|^2}{\|\mu_k - x_k\|} = \xi\frac{\|\alpha_k d_k\|^2}{\|\mu_k - x_k\|} \leq \frac{F(\mu_k)^\top(x_k - \mu_k)}{\|\mu_k - x_k\|} \leq \frac{F_k^\top(x_k - \mu_k)}{\|\mu_k - x_k\|} \leq \|F_k\|,$$

that is

$$\|\mu_k - x_k\| \leq \frac{L\varepsilon}{\xi}. \tag{37}$$

Using (33) and (37), we have

$$\|\mu_k - \bar{x}\| \leq \|\mu_k - x_k\| + \|x_k - \bar{x}\| \leq \|\mu_k - x_k\| + \|x_0 - \bar{x}\| \leq \left(\frac{L}{\xi} + 1\right)\varepsilon.$$

Based on the Lipschitz continuity of F, it follows that

$$\|F(\mu_k)\| = \|F(\mu_k) - F(\bar{x})\| \leq L\|\mu_k - \bar{x}\| \leq \left(\frac{L}{\xi} + 1\right) L\varepsilon. \tag{38}$$

By (26) and (38), and the line search condition (25), we can obtain

$$\begin{aligned} \|x_{k+1} - x_k\| &= \left\| \frac{-F(\mu_k)^\top (x_k - \mu_k)}{\|F(\mu_k)\|^2} F(\mu_k) \right\| = \frac{F(\mu_k)^\top (x_k - \mu_k)}{\|F(\mu_k)\|} \geq \frac{\xi \alpha_k^2 \|d_k\|^2}{\|F(\mu_k)\|} \\ &\geq \frac{\xi^2}{(L+\xi)L\varepsilon} \alpha_k^2 \|d_k\|^2. \end{aligned} \tag{39}$$

It follows from (35) and (39) that the equality (31) holds.

□

Lemma 7. *Based on Propositions 1 and 2, if $\{\mu_k\}$ and $\{x_k\}$ are sequences generated by (19) and (26) in Algorithm 1, then*

$$\alpha_k \geq \frac{\delta \|F_k\|^2}{(L+\xi)\|d_k\|^2}. \tag{40}$$

Proof. Suppose $\alpha_k \neq \delta$, then $\alpha_k^{'} = \frac{\alpha_k}{\delta}$ does not satisfy (25), that is

$$-F(x_k + \alpha_k^{'} d_k)^\top d_k < \xi \alpha_k^{'} \|d_k\|^2.$$

The inequality above combined with (23) and Proposition 1 yields

$$\begin{aligned} \|F_k\|^2 &= -F_k^\top d_k = (F(x_k + \alpha_k^{'} d_k) - F_k)^\top d_k - F(x_k + \alpha_k^{'} d_k)^\top d_k \\ &\leq L\alpha_k^{'} \|d_k\|^2 + \xi \alpha_k^{'} \|d_k\|^2 \\ &= \frac{L+\xi}{\delta} \alpha_k \|d_k\|^2. \end{aligned}$$

Therefore,

$$\alpha_k \geq \frac{\delta \|F_k\|^2}{(L+\xi)\|d_k\|^2},$$

which completes this proof. □

Lemma 8. *Based on Propositions 1 and 2, if d_k is defined by (20)–(22), then for all $k \geq 0$, the following inequality*

$$\|d_k\| \leq N \tag{41}$$

holds, where $N = L(\varepsilon + 4\sigma c)$.

Proof. It is easy to see that for $k = 0$,

$$\|d_0\| = \|F_0\| \leq L\varepsilon.$$

Now, for $k \geq 1$, from (20)–(22), (34) and (36),

$$\begin{aligned}\|d_k\| =& \left\| -F_k + \beta_k d_{k-1} - \beta_k \frac{F_k^T d_{k-1} F_k}{\|F_k\|^2} \right\| \\ \leq& \|F_k\| + \left\| \beta_k d_{k-1} \| + \| \beta_k \frac{F_k^T d_{k-1} F_k}{\|F_k\|^2} \right\| \\ \leq& \|F_k\| + 2\|\beta_k d_{k-1}\| \\ \leq& \|F_k\| + 2\sigma \|F_{k+1} - F_k\| \\ \leq& \|F_k\| + 2L\sigma \|x_{k+1} - x_k\| \\ \leq& \|F_k\| + 2L\sigma (\|x_{k+1}\| + \|x_k\|) \\ \leq& L\varepsilon + 4L\sigma c.\end{aligned}$$

By setting $N := L(\varepsilon + 4\sigma c)$, the assertion then follows immediately. □

Theorem 1. *Based on Propositions 1 and 2, if $\{x_k\}$ is the sequence generated by (19) and (26) in Algorithm 1, then*

$$\liminf_{k \to \infty} \|F_k\| = 0. \tag{42}$$

Proof. Suppose $\liminf_{k \to \infty} \|F_k\| \neq 0$, then $\nu > 0$ exists such that $\|F_k\| \geq \nu$. Using the sufficient descent condition (23) and the Cauchy–Schwarz inequality, we have

$$\|F_k\|^2 = -F_k^T d_k \leq \|F_k\| \|d_k\|.$$

This implies that $\|d_k\| \geq \nu$. Therefore, it follows from (31) that

$$\lim_{k \to \infty} \alpha_k = 0. \tag{43}$$

However, Lemma 7 shows that

$$\alpha_k \geq \max\left\{ \delta, \frac{\delta \|F_k\|^2}{(L+\xi)\|d_k\|^2} \right\} \geq \max\left\{ \delta, \frac{\delta \nu^2}{(L+\xi)N^2} \right\} > 0.$$

This contradicts (43) and hence $\liminf_{k \to \infty} \|F_k\| = 0$. □

Lemma 9 ([25])**.** *If all singular values of A exceed 1, then SOCAVEs (1) has a unique solution.*

Theorem 2. *Based on Propositions 1 and 3, then $\{x_k\}$ generated by (19) and (26) in Algorithm 1 converges to the unique solution of SOCAVEs (1).*

Proof. Based on Lemma 9 and Proposition 3, it is evident that SOCAVEs (1) has a unique solution. The assertion follows immediately on account of Theorem 1, thereby completing this proof. □

5. Numerical Experiments

In this section, we report some numerical examples to compare the MSCG method with the modified multivariate spectral gradient algorithm proposed by [23] (for short MMSGA), the modified SOR-like method proposed by [28] (for short MSOR), and the generalized Newton method proposed by [4] (for short GN). All codes are written in MATLAB R2020a on a personal computer with Intel(R) CPU of 2.10 GHZ and RAM of 16.00 GB.

The parameters used for the implementation of the MSCG method are chosen to be $\xi = 0.001$, $\delta = 0.5$, and $\sigma = 2$. We choose the parameters of the MMSGA as follows:

$$\beta = 0.5, \quad \sigma = 0.01, \quad r = 0.1, \quad \tau = 0.001, \quad \delta = 0.01.$$

As for the MSOR method, we choose the approximate optimal parameters, which satisfy

$$\omega_{\text{aopt}} = \frac{2}{2\sqrt{\mu_{\min}\mu_{\max}} + 1}$$

and

$$\alpha_{\text{aopt}} = \frac{4\mu_{\min}\mu_{\max} + 2\sqrt{\mu_{\min}\mu_{\max}}}{(\sqrt{\mu_{\max}} + \sqrt{\mu_{\min}})^2},$$

where $\mu_{\min} = 1 - \|A^{-1}\|$ and $\mu_{\max} = 1 + \|A^{-1}\|$. In addition, we choose all parameters of the GN method to be the same as those in [4]. The termination criterion used for the three algorithms is $\|F_k\| \leq 10^{-6}$ or if the number of iterations exceeds 1000. Additionally, we choose $x_0 = (0, \cdots, 0)^\top$ as the initial point.

In our experiments, the performance indicators of the algorithms are shown in Table 1.

Table 1. Performance indicators of the algorithms.

Indicators	Meaning of Representation
Iter	the total number of iterations
Time	the elapsed CPU time in seconds
Res	the norm of absolute residual vectors
Err	the norm of absolute error vectors

Here, the Res and Err are defined as

$$\text{Res} := \|Ax_k - |x_k| - b\|$$

and

$$\text{Err} := \|x_k - \bar{x}\|,$$

where $\bar{x}$ is the exact solution.

Example 1. *Consider SOCAVEs* (1) *with*

$$A = tridiag(-1, 8, -1) \in \mathbb{R}^{n\times n}, \quad \bar{x} = (-1, 1, -1, 1, \cdots, -1, 1)^\top \in \mathbb{R}^n,$$

and $b = A\bar{x} - |\bar{x}|$.

From Table 2, it is evident that these iteration methods provide approximations to the desired solutions for different dimensions of n. The elapsed CPU time of the MSCG method is the shortest among the four algorithms considered. Moreover, Table 2 demonstrates that the elapsed CPU time of the MSCG method is not significantly affected by changes in problem size compared to the other three algorithms.

Example 2. *Let m be a given positive integer and $n = m^2$. We consider SOCAVEs* (1)*, where $A \in \mathbb{R}^{n\times n}$ is expressed as $A = \hat{A} + 4I$ and $b \in \mathbb{R}^n$ is given by $b = A\bar{x} - |\bar{x}|$, where*

$$\bar{x} = (-1, 1, -1, 1, \cdots, -1, 1)^\top, \quad \hat{A} = \begin{pmatrix} B & -I & O & \cdots & O & O \\ -I & B & -I & \cdots & O & O \\ O & -I & B & \cdots & O & O \\ \vdots & \vdots & & \ddots & \vdots & \vdots \\ O & O & \cdots & \cdots & B & -I \\ O & O & \cdots & \cdots & -I & B \end{pmatrix}$$

with $B = tridiag(-1, 4, -1) \in \mathbb{R}^{m\times m}$, $I \in \mathbb{R}^{m\times m}$ *being the identity matrix, and* $O \in \mathbb{R}^{m\times m}$ *being the zero matrix.*

In Example 2, we consider the second-order cone as $\mathcal{K} = \mathcal{K}^{n_1} \times \cdots \times \mathcal{K}^{n_r}$, where $n_1 + \cdots + n_r = n$ and $r \geq 1$. We assume that $n_1 = \cdots = n_r = n/r$. The numerical results are presented in Tables 3–5, demonstrating that all approaches provide approximations to the desired solutions. From Tables 3 to 5, it is evident that the MSCG method, MSOR method, and GN method can effectively and accurately solve the problem, with the GN method exhibiting the highest accuracy among these algorithms. Although the MSCG method has the highest number of iterations, its elapsed CPU time is the shortest.

Table 2. Numerical results for Example 1.

	n	**1000**	**2000**	**3000**	**4000**	**5000**	**6000**
MSCG	Iter	22	23	23	23	23	23
	Time	0.0054	0.0040	0.0023	0.0043	0.0052	0.0048
	Res	7.3344×10^{-7}	4.2396×10^{-7}	5.1559×10^{-7}	5.9285×10^{-7}	6.6094×10^{-7}	7.2249×10^{-7}
	Err	7.3134×10^{-8}	4.2539×10^{-8}	5.1713×10^{-8}	5.9446×10^{-8}	6.6258×10^{-8}	7.2416×10^{-8}
MMSGA	Iter	52	56	56	69	56	59
	Time	0.1355	0.5000	1.1071	2.3687	2.9812	4.4165
	Res	9.6533×10^{-7}	5.7346×10^{-7}	5.9639×10^{-7}	9.4671×10^{-7}	8.0882×10^{-7}	6.9092×10^{-7}
	Err	1.0413×10^{-7}	6.2386×10^{-8}	6.4498×10^{-8}	9.8842×10^{-8}	8.5702×10^{-8}	7.2622×10^{-8}
MSOR	Iter	8	8	8	9	9	9
	Time	0.6787	2.9710	8.1707	19.8087	39.6925	68.8103
	Res	5.3711×10^{-7}	7.6161×10^{-7}	9.3390×10^{-7}	9.2454×10^{-8}	1.0296×10^{-7}	1.1246×10^{-7}
	Err	6.8543×10^{-8}	9.7461×10^{-8}	1.1965×10^{-7}	1.0051×10^{-8}	1.1193×10^{-8}	1.2226×10^{-8}
GN	Iter	4	4	4	4	4	4
	Time	0.0876	0.3075	0.8729	1.7540	2.8281	4.9972
	Res	9.5889×10^{-12}	2.5840×10^{-11}	4.8558×10^{-11}	7.4836×10^{-11}	1.0435×10^{-10}	1.3690×10^{-10}
	Err	1.2270×10^{-12}	3.2606×10^{-12}	6.1695×10^{-12}	9.5089×10^{-12}	1.3127×10^{-11}	1.7491×10^{-11}

Table 3. Numerical results for Example 2.

	m	**20**	**30**	**40**	**50**	**60**	**70**
MSCG	Iter	26	27	27	27	27	28
	Time	0.0092	0.0071	0.0017	0.0027	0.0052	0.0057
	Res	4.3855×10^{-7}	4.0944×10^{-7}	4.7299×10^{-7}	6.6179×10^{-7}	9.7673×10^{-7}	4.8759×10^{-7}
	Err	3.7474×10^{-8}	3.4829×10^{-8}	4.0059×10^{-8}	5.6225×10^{-8}	8.3402×10^{-8}	4.1533×10^{-8}
MMSGA	Iter	97	125	129	133	148	177
	Time	0.0734	0.2875	0.8148	1.9196	4.2102	9.6016
	Res	9.8940×10^{-7}	9.6225×10^{-7}	8.9834×10^{-7}	9.5188×10^{-7}	9.8214×10^{-7}	7.2742×10^{-7}
	Err	1.8880×10^{-7}	1.8361×10^{-7}	1.5265×10^{-7}	1.7421×10^{-7}	1.8667×10^{-7}	1.3863×10^{-7}
MSOR	Iter	10	10	10	10	10	10
	Time	0.0815	0.0634	0.2328	1.9189	10.3479	33.3332
	Res	1.0685×10^{-8}	2.7920×10^{-8}	4.4075×10^{-8}	5.9449×10^{-8}	7.4367×10^{-8}	8.9001×10^{-8}
	Err	1.5480×10^{-9}	3.9014×10^{-9}	6.1004×10^{-9}	8.1958×10^{-9}	1.0231×10^{-8}	1.2228×10^{-8}
GN	Iter	4	4	4	4	4	4
	Time	0.0508	0.0631	0.1839	0.5338	1.5809	3.1439
	Res	2.4441×10^{-13}	1.7123×10^{-12}	5.8799×10^{-12}	1.4538×10^{-11}	2.5154×10^{-11}	4.2559×10^{-11}
	Err	3.0969×10^{-14}	2.1237×10^{-13}	7.4333×10^{-13}	1.8532×10^{-12}	3.2322×10^{-12}	5.4819×10^{-12}

In Example 2, A is a sparse block tridiagonal matrix, and $A - I$ is a symmetric positive matrix. Therefore, the convergent condition of the MSCG method is satisfied. The numerical results are listed in Table 3. It can be seen from Table 3 that the MSCG method is the most effective among the four algorithms. In addition, the GN method is the most accurate.

Example 3. *Consider SOCAVEs* (1)*; we choose* $A \in \mathbb{R}^{n\times n}$ *and* $b \in \mathbb{R}^n$ *by* $A = \hat{A} + 4I$ *and* $b = A\bar{x} - |\bar{x}|$*, respectively. Here,* $\hat{A}$ *is defined as follows*

$$\hat{A} = \begin{pmatrix} B & -0.5I & O & \cdots & O & O \\ -1.5I & B & -0.5I & \cdots & O & O \\ O & -1.5I & B & \cdots & O & O \\ \vdots & \vdots & & \ddots & \vdots & \vdots \\ O & O & \cdots & \cdots & B & -0.5I \\ O & O & \cdots & \cdots & -1.5I & B \end{pmatrix}$$

with $B = tridiag(-1.5, 4, -0.5) \in \mathbb{R}^{m\times m}$, $\bar{x} = (1, 2, 1, 2, \cdots, 1, 2)^\top$.

Table 4. Numerical results for Example 3.

	m	**20**	**30**	**40**	**50**	**60**	**70**
MSCG	Iter	100	102	104	106	106	108
	Time	0.0547	0.0067	0.0058	0.0090	0.0145	0.0204
	Res	7.7706×10^{-7}	8.7960×10^{-7}	8.3790×10^{-7}	7.3681×10^{-7}	8.7462×10^{-7}	7.1258×10^{-7}
	Err	1.4827×10^{-7}	1.6905×10^{-7}	1.6158×10^{-7}	1.4236×10^{-7}	1.6921×10^{-7}	1.3799×10^{-7}
MMSGA	Iter	282	279	300	413	532	517
	Time	0.1903	0.6642	1.8153	5.9778	16.6377	30.1169
	Res	7.5264×10^{-7}	7.6801×10^{-7}	9.3640×10^{-7}	9.6159×10^{-7}	9.5593×10^{-7}	9.7385×10^{-7}
	Err	1.5684×10^{-7}	1.6204×10^{-7}	2.0482×10^{-7}	2.0997×10^{-7}	2.1293×10^{-7}	2.2972×10^{-7}
MSOR	Iter	9	10	10	10	10	10
	Time	0.3237	0.1138	0.3903	2.7686	12.8657	37.7472
	Res	5.2065×10^{-7}	1.6860×10^{-7}	2.1851×10^{-7}	2.6716×10^{-7}	3.1549×10^{-7}	3.6379×10^{-7}
	Err	9.9605×10^{-8}	2.9425×10^{-8}	4.0864×10^{-8}	5.2339×10^{-8}	6.3860×10^{-8}	7.5418×10^{-8}
GN	Iter	3	4	4	4	4	4
	Time	0.0739	0.0673	0.1877	0.6389	1.6707	3.1735
	Res	8.5492×10^{-7}	1.5275×10^{-12}	4.9211×10^{-12}	1.1935×10^{-11}	2.3194×10^{-11}	3.7757×10^{-11}
	Err	1.3707×10^{-7}	2.0417×10^{-13}	6.5438×10^{-13}	1.5791×10^{-12}	3.0854×10^{-12}	5.0251×10^{-12}

In contrast to Example 2, $A - I$ is an asymmetric positive matrix, and the numerical results for this scenario are presented in Table 3. It shows that the MSCG method exhibits superior computing performance compared to the other three algorithms.

Example 4. *Consider SOCAVEs* (1) *which is generated the same as Example 1 with* $n = 1000$*. But, here the SOCs is given by* $\mathcal{K}^n := \mathcal{K}^{n_1} \times \cdots \times \mathcal{K}^{n_r}$*, where* $n_1 = \cdots = n_r = \frac{n}{r}$*.*

In Example 4, we choose that $r = 2, 5, 10, 20, 50, 100$. The numerical results are presented in Table 5, demonstrating that all approaches provide approximations to the desired solutions. It is evident that the MSCG method, MMSGA method, MSOR method, and GN method can effectively and accurately solve the problem, with the GN method exhibiting the highest accuracy among these algorithms. Although the MSCG method has a higher iteration count compared to the MSOR method and GN method, its elapsed CPU time is the shortest.

Example 5. *This example is an adaptation of Example* 1 *from the work of [23]. We select a random A according to the following structure:*

$$A = round(4 * eye(n, n) - 0.02 * (2 * rand(n, n) - 1)),$$

then we choose a random $\bar{x} \in \mathbb{R}^n$*, and subsequently, we compute* $b = A\bar{x} - |\bar{x}|$*. Finally, we randomly generate* $x_0 \in \mathbb{R}^n$ *in the interval* $[-5, 5]$ *as the initial point for the iterative process.*

We analyze the numerical comparison among Algorithm 1, MMSGA, MSOR, and GN. In particular, we utilize the performance profile introduced in [36] as a method of evaluation and the performance measurement is the running CPU time.

Table 5. Numerical results for Example 4.

	r	**2**	**5**	**10**	**20**	**50**	**100**
MSCG	Iter	23	23	23	23	23	23
	Time	0.0069	0.0045	0.0028	0.0044	0.0100	0.0159
	Res	6.6105×10^{-7}	6.6187×10^{-7}	6.4937×10^{-7}	6.4864×10^{-7}	6.6740×10^{-7}	6.5931×10^{-7}
	Err	6.2687×10^{-8}	6.2658×10^{-8}	6.1515×10^{-8}	6.1365×10^{-8}	6.2760×10^{-8}	6.1137×10^{-8}
MMSGA	Iter	67	63	77	78	67	66
	Time	0.1772	0.1563	0.1930	0.2027	0.1836	0.1914
	Res	7.9526×10^{-7}	9.9459×10^{-7}	7.3965×10^{-7}	9.6005×10^{-7}	9.9015×10^{-7}	6.0634×10^{-7}
	Err	8.3993×10^{-8}	1.0422×10^{-7}	8.0545×10^{-8}	1.0009×10^{-7}	1.0606×10^{-7}	7.0451×10^{-8}
MSOR	Iter	8	8	8	8	8	8
	Time	0.7350	0.5994	0.5857	0.4541	0.6731	0.6574
	Res	5.4330×10^{-7}	5.4816×10^{-7}	5.5486×10^{-7}	5.6886×10^{-7}	6.1015×10^{-7}	6.6834×10^{-7}
	Err	7.0453×10^{-8}	7.0641×10^{-8}	6.9921×10^{-8}	6.8771×10^{-8}	6.7298×10^{-8}	6.7406×10^{-8}
GN	Iter	4	4	4	4	4	4
	Time	0.0957	0.0646	0.0705	0.0914	0.1561	0.2753
	Res	4.6026×10^{-12}	1.6831×10^{-12}	8.0434×10^{-13}	3.4623×10^{-13}	7.9322×10^{-14}	5.7396×10^{-14}
	Err	5.8097×10^{-13}	2.1482×10^{-13}	1.0123×10^{-13}	4.3809×10^{-14}	9.9165×10^{-15}	7.3165×10^{-15}

Figure 1 illustrates the performance profile of the computing time in Algorithm 1 in the range of $\tau \in [1, 1487]$ for four solvers on 200 randomly generated test problems from Example 5. From Figure 1, it is evident that the proposed methods outperform MMSGA, MSOR, and GN nearly 100% in CPU time. This demonstrates the superior performance and advantages of the proposed methods over the other algorithms.

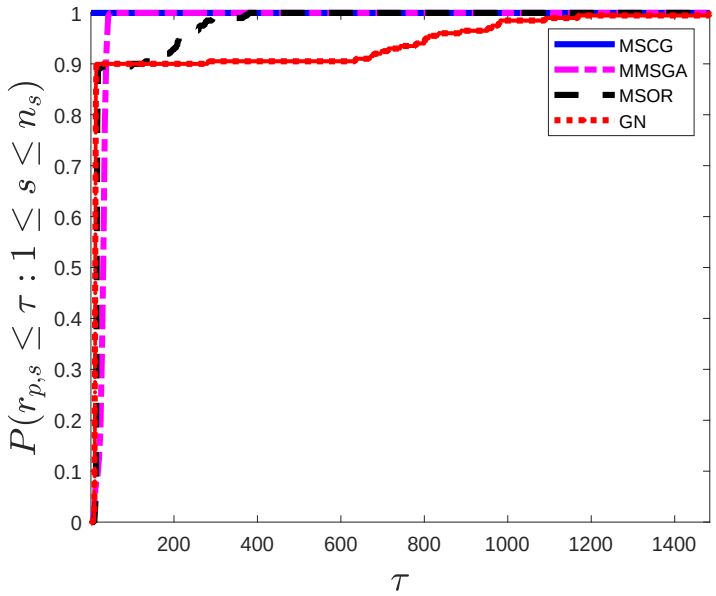


Figure 1. Performance profile of computation time of four different algorithms.

Indeed, for large-scale sparse problems, our proposed method is much more efficient than three existing algorithms. However, its numerical performance deteriorates when dealing with dense coefficient matrices.

6. Conclusions

A modified spectral conjugate gradient (MSCG) method is developed for solving SOCAVEs (1). The convergence results of the MSCG method are proved under certain assumptions. Some numerical results of the MSCG method for solving SOCAVEs (1) demonstrate the effectiveness of the proposed approach. There are also some issues to be further studied, including the following: (i) The necessary and sufficient conditions for solving SOCAVEs are worth further research. (ii) Dynamic models have demonstrated significant success in addressing AVEs, as evidenced by works such as [37–39]. Consequently, how the dynamic models solve SOCAVEs deserves further investigation.

Author Contributions: Conceptualization, guidance, methodology, editing and revision, L.G.; editing, review and revision, original draft preparation, Z.L.; editing and review, software, validation, J.Z.; original draft preparation, review and revision, software, validation, Z.W. All authors have read and agreed to the published version of the manuscript.

Funding: Supported partially by the National Natural Science Foundation of China (Grant No. 12201275), the Ministry of Education in China of Humanities and Social Science Project (Grant No. 21YJCZH204) and the Liaoning Provincial Department of Education (Grant No. JYTZD2023072).

Data Availability Statement: Data are contained within the article.

Acknowledgments: The authors express sincere gratitude to the referees for their valuable comments and suggestions, which greatly contributed to the improvement of this paper.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Chen, J.S.; Tseng, P. An unconstrained smooth minimization reformulation of the second-order cone complementarity problem. *Math. Program.* **2005**, *104*, 293–327. [CrossRef]
2. Chen, J.S.; Pan, S.H. A survey on SOC complementarity functions and solution methods for SOCPs and SOCCPs. *Pac. J. Optim.* **2012**, *8*, 33–74.
3. Fukushima, M.; Luo, Z.Q.; Tseng, P. Smoothing functions for second-order-cone complementarity problems. *SIAM J. Optim.* **2002**, *12*, 436–460. [CrossRef]
4. Hu, S.L.; Huang, Z.H.; Zhang, Q. A generalized newton method for absolute value equations associated with second order cones. *J. Comput. Appl. Math.* **2011**, *235*, 1490–1501. [CrossRef]
5. Ke, Y.F.; Ma, C.F.; Zhang, H. The modulus-based matrix splitting iteration methods for second-order cone linear complementarity problems. *Numer. Algorithm* **2018**, *79*, 1283–1303. [CrossRef]
6. Miao, X.H.; Yang, J.T.; Saheya, B.; Chen, J.S. A smoothing newton method for absolute value equation associated with second-order cone. *Appl. Numer. Math.* **2017**, *120*, 82–96. [CrossRef]
7. Miao, X.H.; Yao, K.; Yang, C.Y.; Chen, J.S. Levenberg-Marquardt method for absolute value equation associated with second-order cone. *Numer. Algebr. Control. Optim.* **2022**, *12*, 47–61. [CrossRef]
8. Nguyen, C.T.; Saheya, B.; Chang, Y.L.; Chen, J.S. Unified smoothing functions for absolute value equation associated with second-order cone. *Appl. Numer. Math.* **2019**, *135*, 206–227. [CrossRef]
9. Rohn, J. A theorem of the alternatives for the equation $Ax + B|x| = b$. *Linear Multilinear Algebra* **2004**, *52*, 421–426. [CrossRef]
10. Mangasarian, O.L. A generalized newton method for absolute value equations. *Optim. Lett.* **2009**, *3*, 101–108. [CrossRef]
11. Prokopyev, O.L. On equivalent reformulations for absolute value equations. *Comput. Optim. Appl.* **2009**, *44*, 363–372. [CrossRef]
12. Hladík, M. Properties of the solution set of absolute value equations and the related matrix classes. *SIAM J. Matrix Anal. Appl.* **2023**, *44*, 175–195. [CrossRef]
13. Mangasarian, O.L.; Meyer, R.R. Absolute value equations. *Linear Algebra Its Appl.* **2006**, *419*, 359–367. [CrossRef]
14. Mezzadri, F. On the solution of general absolute value equations. *Appl. Math. Lett.* **2020**, *107*, 106462. [CrossRef]
15. Wu, S.L.; Li, C.X. The unique solution of the absolute value equations. *Appl. Math. Lett.* **2018**, *76*, 195–200. [CrossRef]
16. Zamani, M.; Hladík, M. Error bounds and a condition number for the absolute value equations. *Math. Program.* **2023**, *198*, 85–113. [CrossRef]
17. Ke, Y.F.; Ma, C.F. SOR-like iteration method for solving absolute value equations. *Appl. Math. Comput.* **2017**, *311*, 195–202. [CrossRef]
18. Guo, P.; Wu, S.L.; Li, C.X. On the SOR-like iteration method for solving absolute value equations. *Appl. Math. Lett.* **2019**, *97*, 107–113. [CrossRef]
19. Zhang, Y.M.; Yu, D.M.; Yuan, Y.F. On the alternative SOR-like iteration method for solving absolute value equations. *Symmetry* **2023**, *15*, 589. [CrossRef]

20. Dong, X.; Shao, X.H.; Shen, H.L. A new SOR-like method for solving absolute value equations. *Appl. Numer. Math.* **2020**, *156*, 410–421. [CrossRef]
21. Ke, Y.F. The new iteration algorithm for absolute value equation. *Appl. Math. Lett.* **2020**, *99*, 105990. [CrossRef]
22. Yu, D.M.; Chen, C.R.; Han, D.R. A modified fixed point iteration method for solving the system of absolute value equations. *Optimization* **2022**, *71*, 449–461. [CrossRef]
23. Yu, Z.S.; Li, L.; Yuan, Y. A modified multivariate spectral gradient algorithm for solving absolute value equations. *Appl. Math. Lett.* **2021**, *121*, 107461. [CrossRef]
24. Rahpeymaii, F.; Amini, K.; RostamyMalkhalifeh, M. A new three-term spectral subgradient method for solving absolute value equation. *Int. J. Comput. Math.* **2023**, *100*, 440–452. [CrossRef]
25. Miao, X.H.; Hsu, W.M.; Nguyen, C.T.; Chen, J.S. The solvabilities of three optimization problems associated with second-order cone. *J. Nonlinear Convex Anal.* **2021**, *22*, 937–967.
26. Miao, X.H.; Chen, J.S. On matrix characterizations for P-property of the linear transformation in second-order cone linear complementarity problems. *Linear Algebra Its Appl.* **2021**, *613*, 271–294. [CrossRef]
27. Yu, D.M.; Wang, Z.W.; Chen, C.R.; Han, D.R. A non-monotone smoothing Newton algorithm for absolute value equations with second-order cone. *Math. Numer. Sin.* **2023**, *45*, 251.
28. Huang, B.H.; Li, W. A modified SOR-like method for absolute value equations associated with second order cones. *J. Comput. Appl. Math.* **2022**, *400*, 113745. [CrossRef]
29. Alizadeh, F.; Goldfarb, D. Second-order cone programming. *Math. Program.* **2003**, *95*, 3–51. [CrossRef]
30. Faraut, J.; Korányi, A. *Analysis on Symmetric Cones*; Oxford University Press: Oxford, UK, 1994.
31. Huang, B.H.; Ma, C.F. Convergent conditions of the generalized newton method for absolute value equation over second order cones. *Appl. Math. Lett.* **2019**, *92*, 151–157. [CrossRef]
32. Liu, J.K.; Feng, Y.M.; Zou, L.M. A spectral conjugate gradient method for solving large-scale unconstrained optimization. *Comput. Math. Appl.* **2019**, *77*, 731–739. [CrossRef]
33. Solodov, M.V.; Svaiter, B.F. A globally convergent inexact newton method for systems of monotone equations. In *Reformulation: Nonsmooth, Piecewise Smooth, Semismooth and Smoothing Methods*; Springer: Boston, MA, USA, 1999; pp. 355–369.
34. Bauschke, H.H.; Combettes, P.L. *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*; Springer: New York, NY, USA, 2011.
35. Bot, R.I.; Csetnek, E.R.; Nguyen, D.K. A proximal minimization algorithm for structured nonconvex and nonsmooth problems. *SIAM J. Optim.* **2019**, 29, 1300–1328. [CrossRef]
36. Dolan, E.D.; Moré, J.J. Benchmarking optimization software with performance profiles. *Math. Program.* **2002**, *91*, 201–213. [CrossRef]
37. Chen, C.R.; Yang, Y.N.; Yu, D.M.; Han, D.R. An inverse-free dynamical system for solving the absolute value equations. *Appl. Numer. Math.* **2021**, *168*, 170–181. [CrossRef]
38. Ju, X.X.; Yang, X.S.; Feng, G.; Che, H.J. Neurodynamic optimization approaches with finite/fixed-time convergence for absolute value equations. *Neural Netw.* **2023**, *165*, 971–981. [CrossRef]
39. Yu, D.M.; Zhang, G.H.; Chen, C.R.; Han, D.R. The neural network models with delays for solving absolute value equations. *Neurocomputing* **2024**, *589*, 127707. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.