A descent method for a reformulation of the second-order cone complementarity problem

Jein-Shan Chen
Department of Mathematics
National Taiwan Normal University
Taipei 11677, Taiwan

Shaohua Pan
School of Mathematical Sciences
South China University of Technology
Guangzhou 510640, China

July 28, 2006
(first revised September 19, 2006)
(second revised January 31, 2007)

Abstract  Analogous to the nonlinear complementarity problem (NCP) and the semi-definite complementarity problem (SDCP), a popular approach to solving the second-order cone complementarity problem (SOCCP) is to reformulate it as an unconstrained minimization of a certain merit function over $\mathbb{R}^n$. In this paper, we present a descent method for solving the unconstrained minimization reformulation of the SOCCP which is based on the Fischer-Burmeister merit function associated with second-order cone [4], and prove its global convergence. Particularly, we compare the numerical performance of the method for the symmetric affine SOCCP generated randomly with the Fischer-Burmeister merit function approach [4]. The comparison results indicate that, if a scaling strategy is imposed on the test problem, the descent method proposed is comparable with the merit function approach in the CPU time for solving test problems although the former may require more function evaluations.

Key words. Second-order cone, complementarity, merit function, descent method

1Member of Mathematics Division, National Center for Theoretical Sciences, Taipei Office. The author’s work is partially supported by National Science Council of Taiwan. E-mail: jschen@math.ntnu.edu.tw, FAX: 886-2-29332342.
2E-mail: shhpan@scut.edu.cn.
1 Introduction

In this paper, we consider the following SOCP of finding $\zeta \in \mathbb{R}^n$ satisfying
\[
\langle F(\zeta), \zeta \rangle = 0, \quad F(\zeta) \in \mathcal{K}, \quad \zeta \in \mathcal{K},
\]
where $\langle \cdot, \cdot \rangle$ is the Euclidean inner product, $F : \mathbb{R}^n \to \mathbb{R}^n$ is a smooth (i.e., continuously differentiable) mapping, and $\mathcal{K}$ is the Cartesian product of second-order cones (SOC), also called Lorentz cones [9]. In other words,
\[
\mathcal{K} = \mathcal{K}^{n_1} \times \cdots \times \mathcal{K}^{n_m},
\]
where $m, n_1, \ldots, n_m \geq 1$, $n_1 + \cdots + n_m = n$, and
\[
\mathcal{K}^{n_i} := \{(x_1, x_2) \in \mathbb{R} \times \mathbb{R}^{n_{i-1}} \mid \|x_2\| \leq x_1\},
\]
with $\| \cdot \|$ denoting the Euclidean norm and $\mathcal{K}^1$ denoting the set of nonnegative reals $\mathbb{R}_+$. A special case of (2) is $\mathcal{K} = \mathbb{R}^n_+$, the nonnegative orthant in $\mathbb{R}^n$, which corresponds to $m = n$ and $n_1 = \cdots = n_m = 1$. If $\mathcal{K} = \mathbb{R}^n_+$, then (1) reduces to the nonlinear complementarity problem (NCP). The NCP plays a fundamental role in optimization theory and has many applications in engineering and economics; see, e.g., [7, 10, 11, 12]. Unless otherwise stated, in the first three sections of this paper, we assume $\mathcal{K} = \mathcal{K}^n$ for simplicity, i.e., $\mathcal{K}$ is a single second-order cone (all the analysis can be carried over to the case where $\mathcal{K}$ is a product of second-order cones without difficulty).

There have been proposed various methods for solving the SOCP. They include interior-point methods [1, 25, 28, 29, 31], reformulating SOC constraints as smooth convex constraints [32], and (non-interior) smoothing Newton methods [6, 16, 20]. These methods require solving a nontrivial system of linear equations at each iteration. In the recent paper [4], an alternative approach based on reformulating the SOCP as an unconstrained smooth minimization problem was studied. In particular, they were finding a smooth function $\psi : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}_+$ such that
\[
\psi(x, y) = 0 \iff \langle x, y \rangle = 0, \quad x \in \mathcal{K}^n, \quad y \in \mathcal{K}^n.
\]
We call such a $\psi$ a merit function. Then SOCP can be expressed as an unconstrained smooth (global) minimization problem:
\[
\min_{\zeta \in \mathbb{R}^n} \psi(F(\zeta), \zeta).
\]
Various gradient methods such as conjugate gradient methods and quasi-Newton methods [2, 15] can be applied to (5). For this approach to be effective, the choice of $\psi$ is crucial. In the case of NCP, a popular choice is
\[
\psi_{\text{FB}}(a, b) = \frac{1}{2} \sum_{i=1}^{n} \phi_{\text{FB}}(a_i, b_i)^2
\]
for all \(a = (a_1, \ldots, a_n)^T \in \mathbb{R}^n\) and \(b = (b_1, \ldots, b_n)^T \in \mathbb{R}^n\), where \(\phi_{FB}\) is the well-known Fischer-Burmeister NCP-function \([13, 14]\) defined by

\[
\phi_{FB}(a_i, b_i) = \sqrt{a_i^2 + b_i^2 - a_i - b_i}.
\]

It has been shown that \(\psi_{FB}\) is smooth (even though \(\phi_{FB}\) is not differentiable) and is a merit function for NCP \([8, 22, 23]\). These two functions can be extended to the case of SOCCP via Jordan algebra shown as below. For any \(x = (x_1, x_2), y = (y_1, y_2) \in \mathbb{R} \times \mathbb{R}^{n-1}\), we define their Jordan product associated with \(K^n\) as

\[
x \circ y := (\langle x, y \rangle, y_1 x_2 + x_1 y_2).
\]

The identity element under this product is \(e := (1, 0, \ldots, 0)^T \in \mathbb{R}^n\). We write \(x^2\) to mean \(x \circ x\) and write \(x + y\) to mean the usual componentwise addition of vectors. It is known that \(x^2 \in K^n\) for all \(x \in \mathbb{R}^n\). Moreover, if \(x \in K^n\), then there exists a unique vector in \(K^n\), denoted by \(x^{1/2}\), such that \((x^{1/2})^2 = x^{1/2} \circ x^{1/2} = x\). Then,

\[
\phi_{FB}(x, y) := (x^2 + y^2)^{1/2} - x - y
\]

is well-defined for all \((x, y) \in \mathbb{R}^n \times \mathbb{R}^n\) and maps \(\mathbb{R}^n \times \mathbb{R}^n\) to \(\mathbb{R}^n\). It was shown in \([16]\) that \(\phi_{FB}(x, y) = 0\) if and only if \(\langle x, y \rangle = 0, x \in K^n, y \in K^n\). Hence, \(\psi_{FB} : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}_+\) given by

\[
\psi_{FB}(x, y) := \frac{1}{2} \|\phi_{FB}(x, y)\|^2,
\]

is a merit function for SOCCP because \(\psi_{FB}\) satisfies (4) as well. Therefore, the SOCCP is equivalent to the global minimization problem:

\[
\min_{\zeta \in \mathbb{R}^n} f_{FB}(\zeta) := \psi_{FB}(F(\zeta), \zeta).
\]

It was also shown in the paper \([4]\) that, like the NCP case, \(\psi_{FB}\) is smooth and, when \(\nabla F\) is positive semi-definite, every stationary point of (8) solves the SOCCP. For SDCP, which is a natural extension of NCP where \(\mathbb{R}^n_+\) is replaced by the cone of positive semi-definite matrices \(S^n_+\) and the partial order \(\leq\) is also changed by \(\preceq_{S^n_+}\) (a partial order associated with \(S^n_+\) where \(A \preceq_{S^n_+} B\) means \(B - A \in S^n_+\)) accordingly, the aforementioned features hold for the following analog of the SDCP merit function studied by Yamashita and Fukushima in \([33]\):

\[
\psi_{YF}(x, y) := \psi_0(\langle x, y \rangle) + \psi_{FB}(x, y),
\]

where \(\psi_0 : \mathbb{R} \to [0, \infty)\) is any smooth function satisfying

\[
\psi_0(t) = 0 \quad \forall t \leq 0 \quad \text{and} \quad \psi'_0(t) > 0 \quad \forall t > 0.
\]

In \([33]\), \(\psi_0(t) = \frac{1}{4}(\max\{0, t\})^4\) was considered. In fact, the function \(\psi_{YF}\), which was recently studied in \([4]\), is also a SOCCP version merit function that enjoys favorable
properties as what $\psi_{FB}$ has. Moreover, $\psi_{VF}$ possesses properties of bounded level sets and error bound.

In this paper, we focus on the following equivalent reformulation of SOCCP, which arises via the merit function $\psi_{VF}$ defined as in (9)-(10):

$$\min_{\zeta \in \mathbb{R}^n} f_{VF}(\zeta) := \psi_{VF}(F(\zeta), \zeta).$$

(11)

We are motivated by the work [33] showing a descent method for the SDCP. Thus, the main purpose of the paper is to explore the extension to SOCCP. In other words, we wish to adopt the algorithm therein to solve the equivalent reformulation (11) of the SOCCP and prove its global convergence (see Sec. 3). In particular, we also compare the numerical performance of the descent algorithm for the symmetric affine SOCCPs generated randomly with the Fischer-Burmeister merit function approach [4]. Here it is worth of pointing out that the proposed algorithm does not work for the other reformulation (8). The reason is that $f_{FB}(\zeta)$ lacks property of bounded level sets and does not provide error bound due to the absence of the term $\psi_0$.

Some words about our notation. Throughout this paper, $\mathbb{R}^n$ denotes the space of $n$-dimensional real column vectors. For any differentiable function $f : \mathbb{R}^n \to \mathbb{R}$, $\nabla f(x)$ denotes the gradient of $f$ at $x$. For any differentiable mapping $F : \mathbb{R}^n \to \mathbb{R}^m$, $\nabla F(x)$ is a $n \times m$ matrix which denotes the transposed Jacobian of $F$ at $x$.

2 Preliminaries

As mentioned in the introduction, $\psi_{VF}$ satisfies (4), so the SOCCP can be recast as an equivalent global minimization (11). It was shown in [4] that the function $f_{VF}$ is smooth, has bounded level sets, and provides error bound for the unconstrained minimization reformulation. Moreover, every stationary point of problem (11) is a solution of the SOCCP. In this section, we review some basic concepts and properties that will be used for proving the convergence results of the descent algorithm later. Since the work of [4] already includes as special cases the following lemmas, we here omit the proofs.

Lemma 2.1 [4, Prop. 3.2] Let $\phi_{FB}, \psi_{FB}$ be given by (6) and (7), respectively, and $\psi_{VF}$ be given by (9)-(10). Then $\psi_{FB}$ and $\psi_{VF}$ are both smooth on $\mathbb{R}^n \times \mathbb{R}^n$.

Lemma 2.2 [4, Prop. 4.2] Let $\psi_{VF}$ be given by (9)-(10) and $f_{VF}(\zeta)$ be defined as (11). Then, for every $\zeta \in \mathbb{R}^n$ such that $\nabla F(\zeta)$ is positive semi-definite, either $f_{VF}(\zeta) = 0$ or $\nabla f_{VF}(\zeta) \neq 0$ with $\langle d(\zeta), \nabla f_{VF}(\zeta) \rangle < 0$, where

$$d(\zeta) := -\left(\psi'_0(F(\zeta), \zeta) + \nabla_x \psi_{FB}(F(\zeta), \zeta)\right).$$

(12)
In what follows, we say that $F$ is monotone if
\[ \langle F(\zeta) - F(\xi), \zeta - \xi \rangle \geq 0 \quad \forall \zeta, \xi \in \mathbb{R}^n \]
and $F$ is strongly monotone if there exists $\rho > 0$ such that
\[ \langle F(\zeta) - F(\xi), \zeta - \xi \rangle \geq \rho \| \zeta - \xi \|_2^2 \quad \forall \zeta, \xi \in \mathbb{R}^n. \]
It is well known that, when $F$ is differentiable, $F$ is monotone if and only if $\nabla F(\zeta)$ is positive semi-definite for all $\zeta \in \mathbb{R}^n$ while $F$ is strongly monotone if and only if $\nabla F(\zeta)$ is positive definite for all $\zeta \in \mathbb{R}^n$.

Lemma 2.3 [4, Prop. 5.2] Suppose that $F$ is a differentiable and monotone mapping from $\mathbb{R}^n$ to $\mathbb{R}^n$. Suppose also that the SOCCP (1) is strictly feasible, i.e., there exists $\hat{\zeta} \in \mathbb{R}^n$ such that $F(\hat{\zeta}), \hat{\zeta} \in \text{int}(\mathcal{K}^n)$. Then the level set
\[ L(\gamma) := \{ \zeta \in \mathbb{R}^n | f_{\text{YF}}(\zeta) \leq \gamma \} \]
is nonempty and bounded for all $\gamma \geq 0$, where $f_{\text{YF}}$ is given by (11).

Remark 2.1 It is known that Lemma 2.3 is also true if the conditions of monotonicity and strict feasibility is replaced by strong monotonicity.

We next recall some basic results about the spectral factorization associated with $\mathcal{K}^n$. For any $x = (x_1, x_2) \in \mathbb{R} \times \mathbb{R}^{n-1}$, it admits a spectral factorization of the form
\[ x = \lambda_1(x) \cdot u_x^{(1)} + \lambda_2(x) \cdot u_x^{(2)}, \tag{13} \]
where $\lambda_i(x)$ and $u_x^{(i)}$ for $i = 1, 2$ are the spectral values and the associated spectral vectors of $x$ given by
\[
\begin{align*}
\lambda_i(x) &= x_1 + (-1)^i \| x_2 \|, \\
u_x^{(i)}(x) &= \begin{cases} 
\frac{1}{2} \left(1, (-1)^i \frac{x_2}{\| x_2 \|}\right) & \text{if } x_2 \neq 0 \\
\frac{1}{2} \left(1, (-1)^i w_2 \right) & \text{if } x_2 = 0
\end{cases} \tag{14}
\end{align*}
\]
with $w_2$ being any vector in $\mathbb{R}^{n-1}$ satisfying $\| w_2 \| = 1$. If $x_2 \neq 0$, the factorization is unique. The set $\{u_x^{(1)}, u_x^{(2)}\}$ is called a Jordan frame and has the following properties.

Property 2.1 For any $x = (x_1, x_2) \in \mathbb{R} \times \mathbb{R}^{n-1}$ with the spectral values $\lambda_1(x), \lambda_2(x)$ and spectral vectors $u_x^{(1)}, u_x^{(2)}$ given as in (14), we have
(a) $u_x^{(1)}$ and $u_x^{(2)}$ are orthogonal under Jordan product and have length $1/\sqrt{2}$, i.e.,
\[ u_x^{(1)} \circ u_x^{(2)} = 0, \quad \| u_x^{(1)} \| = \| u_x^{(2)} \| = \frac{1}{\sqrt{2}}. \]
(b) $u_x^{(1)}$ and $u_x^{(2)}$ are idempotent under Jordan product, i.e. $u_x^{(i)} \circ u_x^{(i)} = u_x^{(i)}$ for $i = 1, 2$.

The spectral factorization (13)-(14) of $x$, as well as $x^2$ and $x^{1/2}$ have various interesting properties; see [16]. For instances, for any $x = (x_1, x_2) \in \mathbb{R} \times \mathbb{R}^{n-1}$, with spectral values $\lambda_1(x), \lambda_2(x)$ and spectral vectors $u_x^{(1)}, u_x^{(2)}$, the following results hold:

1. $x^2 = \lambda_1^2(x) \; u_x^{(1)} + \lambda_2^2(x) \; u_x^{(2)} \in \mathcal{K}^n$.
2. If $x \in \mathcal{K}^n$, then $0 \leq \lambda_1(x) \leq \lambda_2(x)$ and $x^{1/2} = \sqrt{\lambda_1(x)} \; u_x^{(1)} + \sqrt{\lambda_2(x)} \; u_x^{(2)}$.

To close this section, we present a property of $\psi_{\text{FB}}$ associated with the spectral value.

**Lemma 2.4** [4, Lemma 9(a)] For any $\{(x^k, y^k)\}_{k=1}^{\infty} \subseteq \mathbb{R}^n \times \mathbb{R}^n$, let $\lambda_1(x^k) \leq \lambda_2(x^k)$ and $\mu_1(y^k) \leq \mu_2(y^k)$ denote the spectral values of $x^k$ and $y^k$, respectively. Then, if $\lambda_1(x^k) \to -\infty$ or $\mu_1(y^k) \to -\infty$, we have $\psi_{\text{FB}}(x^k, y^k) \to \infty$.

### 3 Main Results

In this section, we propose a descent method for solving the unconstrained minimization reformulation (11) of the SOCCP and prove its global convergence. The proposed method uses $d(\zeta)$ defined as (12) as its direction. Now let us describe the algorithm.

**Algorithm 3.1:**

**Step 0** Choose $\zeta^0 \in \mathbb{R}^n$, $\varepsilon \geq 0$, $\sigma \in (0, 1/2)$, $\beta \in (0, 1)$ and set $k := 0$.

**Step 1** If $f_{\lambda}(\zeta^k) \leq \varepsilon$, then stop.

**Step 2** Compute $d(\zeta^k) := -\left(\psi'_0((F(\zeta^k), \zeta^k))\zeta^k + \nabla_x \psi_{\text{FB}}(F(\zeta^k), \zeta^k)\right)$.

**Step 3** Find a step-size $t_k := \beta^m_k$, where $m_k$ is the smallest nonnegative integer $m$ satisfying the Armijo’s rule:

$$f_{\lambda}(\zeta^k + \beta^m(\zeta^k)) \leq (1 - \sigma \beta^{2m}) f_{\lambda}(\zeta^k). \quad (15)$$

**Step 4** Set $\zeta^{k+1} := \zeta^k + t_k \; d(\zeta^k)$, $k := k + 1$ and go to Step 1.

Note that the above algorithm is $\nabla F$-free, i.e., there is no need to compute the Jacobian matrix of $F$, and moreover, the computation work in each iteration is very small, i.e., only several vector multiplications. In fact, this type of algorithm was also studied for the NCP (see [17]) and the SDCP (see [33]) and the most remarkable feature of this
type of algorithm is that not only the step-size but also the search direction itself is adjusted via the Armijo’s rule. In practical experience, $\sigma$ is usually chosen close to zero, and $\beta$ is usually chosen in $(\frac{1}{10}, 1)$ depending on the confidence we have on the quality of the initial step-size (see [2]).

Next, we prove the global convergence of Algorithm 3.1. Without any loss of generality, we suppose $\varepsilon = 0$ so that the algorithm generates an infinite sequence $\{\zeta_k\}$.

**Proposition 3.1** Suppose that $F$ is monotone and the SOCCP (1) is strictly feasible. Then the sequence $\{\zeta_k\}$ generated by Algorithm 3.1 has at least one accumulation point, and any accumulation point is a solution of the SOCCP (1).

**Proof.** The proof is standard and can be found in [2]. For completeness, we here present its proof by the following three steps.

(i) First, we show that, whenever $\zeta^k$ is not a solution, there exists a nonnegative integer $m_k$ in Step 3 of Algorithm 3.1. Suppose not, then for any positive integer $m$, we have

$$f_{YF}(\zeta^k + \beta^m d(\zeta^k)) - f_{YF}(\zeta^k) > -\sigma \beta^m f_{YF}(\zeta^k).$$

Dividing by $\beta^m$ on both sides and letting $m \to \infty$ yields

$$\langle \nabla f_{YF}(\zeta^k), d(\zeta^k) \rangle \geq 0.$$

Since $F$ is monotone which is equivalent to $\nabla F(\zeta)$ is positive semi-definite, the inequality (16) contradicts Lemma 2.2. Hence, we can find an integer $m_k$ in Step 3.

(ii) Secondly, we show that the sequence $\{\zeta^k\}$ generated by the algorithm has at least one accumulation point. By the descent property of Algorithm 3.1, the sequence $\{f_{YF}(\zeta_k)\}_{k \in \mathbb{N}}$ is decreasing. Hence by Lemma 2.3, we have that $\{\zeta^k\}$ is bounded, and consequently has at least one accumulation point.

(iii) Finally, we prove that any accumulation point of $\{\zeta^k\}$ is a solution of the SOCCP (1). Let $\zeta^*$ be an arbitrary accumulation point of $\{\zeta^k\}_{k \in \mathbb{N}}$. In other words, there is a subsequence $\{\zeta^k\}_{k \in K}$ converging to $\zeta^*$, where $K$ is a subset of $N$. We know $d(\cdot)$ is continuous (since $\psi_0$ and $\psi_{FB}$ are smooth) which implies $\{d(\zeta^k)\}_{k \in K}$ converges to $d(\zeta^*)$.

Next, we need to discuss two cases. First, we consider the case where there exists a constant $\bar{\beta}$ such that $\beta^m_k \geq \bar{\beta} > 0$ for all $k \in K$. Then, from (15), we have

$$f_{YF}(\zeta^{k+1}) \leq (1 - \sigma \bar{\beta}^2) f_{YF}(\zeta^k)$$

for all $k \in K$ and the entire sequence $\{f_{YF}(\zeta^k)\}_{k \in K}$ is decreasing. Thus, we obtain $f_{YF}(\zeta^*) = 0$ (by taking the limit) which says $\zeta^*$ is a solution of the SOCCP (1). Now, we consider the other case where there exists a further subsequence such that $\beta^m_k \to 0$. Note that by Armijo’s rule (15) in Step 3, we have

$$f_{YF}(\zeta^k + \beta^{m_k-1} d(\zeta^k)) - f_{YF}(\zeta^k) > -\sigma \beta^{2(m_k-1)} f_{YF}(\zeta^k).$$
Dividing by $\beta^{m_k-1}$ both sides and passing the limit on the further subsequence, we obtain
\[
\langle \nabla f_{\nu_F}(\zeta^*), d(\zeta^*) \rangle \geq 0,
\]
which yields that $\zeta^*$ is a solution of the SOCCP (1) by Lemma 2.2. 

**Proposition 3.2** Let $F$ be a continuously differentiable and strongly monotone mapping. Then the sequence $\{\zeta^k\}$ generated by Algorithm 3.1 converges to the unique solution of the SOCCP (1).

**Proof.** The proof is routine (see [7]), however, we present it for completeness. We know that the property of bounded level sets is also held when $F$ is strongly monotone, so following the same arguments as in the proof of Prop. 3.1, we again obtain that $\{\zeta^k\}$ has at least one accumulation point and any accumulation point is a solution of the SOCCP (1).

On the other hand, the strong monotonicity of $F$ implies that the SOCCP (1) has at most one solution. To see this, say there are two solutions $\zeta^*, \xi^* \in \mathbb{R}^n$ such that
\[
\{ \langle F(\zeta^*), \zeta^* \rangle = 0, \\
F(\zeta^*) \in K^n, \quad \zeta^* \in K^n \}
\quad \text{and} \quad
\{ \langle F(\xi^*), \xi^* \rangle = 0, \\
F(\xi^*) \in K^n, \quad \xi^* \in K^n \}.
\]
Since $F$ is strongly monotone, we have $\langle F(\zeta^*) - F(\xi^*), \zeta^* - \xi^* \rangle > 0$. However,
\[
\langle F(\zeta^*) - F(\xi^*), \zeta^* - \xi^* \rangle = \langle F(\zeta^*), \zeta^* \rangle + \langle F(\xi^*), \xi^* \rangle - \langle F(\zeta^*), \xi^* \rangle - \langle F(\xi^*), \zeta^* \rangle
\leq 0,
\]
where the inequality is due to $F(\zeta^*), \zeta^*, F(\xi^*), \xi^*$ are all in $K$. Hence, it is a contradiction and therefore there is at most one solution for the SOCCP (1).

From all the above, it says there is a unique solution $\zeta^*$, so the entire sequence $\{x^k\}$ must converge to $\zeta^*$. 

Prop. 3.1-3.2 may not be so surprising since they seems as expected. Nonetheless, we do not take them for granted before we prove them even though we think they should be true. Now, the results of Prop. 3.1-3.2 do fill up the gap in the literature. We notice that Lemma 2.3 plays an important role in the proofs for them. In fact, the assumption of strict feasibility is necessary for Lemma 2.3 to be held. For example, when $F(\zeta) \equiv 0$, every $\zeta \in K^n$ is a solution of SOCCP (1) and hence the solution set is unbounded. In the following, we continue a further study of considering another (weaker) condition to replace this kind of “strict” condition by $B$ being a $R_0$-function (will be defined in Def. 3.1) that is a new concept recently developed for linear and nonlinear transformations on Euclidean Algebra [19, 26, 30].
Definition 3.1 For a mapping $F : \mathbb{R}^n \to \mathbb{R}^n$, it is called a
(a) $R_{01}$-function if for any sequence $\{\zeta^k\}$ such that
\[ \|\zeta^k\| \to \infty, \quad \frac{(-\zeta^k)_+}{\|\zeta^k\|} \to 0, \quad \frac{(-F(\zeta^k))_+}{\|\zeta^k\|} \to 0, \] (17)
we have
\[ \liminf_{k \to \infty} \frac{\langle \zeta^k, F(\zeta^k) \rangle}{\|\zeta^k\|} > 0; \]
(b) $R_{02}$-function if for any sequence $\{\zeta^k\}$ such that (17), we have
\[ \liminf_{k \to \infty} \frac{\omega(\zeta^k \circ F(\zeta^k))}{\|\zeta^k\|^2} > 0. \]

The above concepts are extensions of the ones defined for NCP and for SDCP. It is also known that every $R_{01}$-function is $R_{02}$-function [24, Lemma 4]; and if $F$ has the uniform Jordan $P$-property (see [19, 26, 30]), then $F$ is $R_{02}$-function [24, Lemma 5]. However, it is not clear whether uniform $P$-property (see [19, 26, 30]) implies $R_{02}$-function or not. With this new concept, Lemma 2.3 and Prop. 3.1 can be improved as Lemma 3.1 and Prop. 3.3, respectively. These results are significant not only they are brand-new but also there is no needs the assumption of strict feasibility therein.

Lemma 3.1 Let $f_{\text{vf}}$ be given as in (11). Suppose that $F$ is a $R_{01}$-function. Then the level set
\[ \mathcal{L}(\gamma) := \{ \zeta \in \mathbb{R}^n \mid f_{\text{vf}}(\zeta) \leq \gamma \} \]
is bounded for all $\gamma \geq 0$.

Proof. We will prove this result by contradiction. Suppose there exists an unbounded sequence $\{\zeta^k\} \subset \mathcal{L}(\gamma)$ for some $\gamma \geq 0$. It can be seen that the sequence of the smaller spectral values of $\{\zeta^k\}$ and $\{F(\zeta^k)\}$ are bounded below. In fact, if not, it follows from Lemma 2.4 that $f_{\text{vf}}(\zeta^k) \to \infty$, which contradicts $\{\zeta^k\} \subset \mathcal{L}(\gamma)$. Therefore, $\{(-\zeta^k)_+\}$ and $\{(-F(\zeta^k))_+\}$ are bounded above, which says the conditions in (17) are satisfied. Then, by the assumption of $R_{01}$-function, we have
\[ \liminf_{k \to \infty} \frac{\langle \zeta^k, F(\zeta^k) \rangle}{\|\zeta^k\|^2} > 0. \]
This yields $\langle \zeta^k, F(\zeta^k) \rangle \to \infty$, and hence $f_{\text{vf}}(\zeta^k) \to \infty$ by definition of $f_{\text{vf}}$ given as in (11). Thus, it is a contradiction to $\{\zeta^k\} \subset \mathcal{L}(\gamma)$.

\[ \square \]
Proposition 3.3 Let $F$ be a continuously differentiable mapping. Suppose that $F$ is $R_{01}$-function. Then the sequence $\{\zeta^k\}$ generated by Algorithm 3.1 has at least one accumulation point, and any accumulation point is a solution of the SOCCP (1).

Proof. By applying Lemma 3.1 and follow the same arguments as in Prop. 3.1, the desired results hold. We omit it. \(\square\)

From [24, 30], the condition of $R_{01}$-function is weaker than strong monotonicity, and it is also weaker than monotonicity plus strict feasibility in certain sense. However, it is not clear yet whether $R_{01}$-function can be replaced by $R_{02}$-function in our brand-new results.

4 Numerical results

In this section, we report our computational experience with solving the symmetric affine SOCCPs generated randomly by the proposed algorithm, and compare the numerical performance with the Fischer-Burmeister merit function approach [4]. Unless otherwise stated, the function $f_{\nu_F}$ in Algorithm 3.1 is always defined as in (11), where $\psi_{\nu_F}$ is defined by (9)-(10) with $\psi_0(t) = \frac{1}{2}(\max\{0, t\})^2$.

The symmetric affine SOCCP is stated as follows: finding $\zeta \in \mathbb{R}^n$ such that

\[
\langle F(\zeta), \zeta \rangle = 0, \quad \zeta \in \mathcal{K}, \quad F(\zeta) = M\zeta + q \in \mathcal{K}
\]  

where $M \in \mathbb{R}^{n \times n}$ and $q \in \mathbb{R}^n$ are a given symmetric positive semidefinite matrix and a vector, respectively. In our experiments, the matrix $M$ and the vector $q$ are generated by the following procedure. Elements of $q$ were chosen randomly from the interval $[-1, 1]$ and the matrix $M$ was obtained by setting $M = NN^T$, where $N$ is a square matrix whose nonzero elements are chosen randomly from the interval $[-1, 1]$. In this procedure, the number of nonzero elements of $N$ is determined so that the nonzero density of $M$ can be approximately estimated.

All experiments were done at a PC with 2.8GHz CPU and 512MB memory. The computer codes were all written in Matlab 6.1. To improve the numerical behavior of Algorithm 3.1, we replaced the standard Armijo-rule by the nonmonotone line search as described in [18], i.e. we computed the smallest nonnegative integer $m$ such that

\[
f_{\nu_F}(\zeta^k + \beta^m d(\zeta^k)) \leq W_k - \sigma \beta^{2m} f_{\nu_F}(\zeta^k)
\]

where $W_k$ is given by

\[
W_k = \max_{j=k-m,\ldots,k} f_{\nu_F}(\zeta^j)
\]
and where, for given nonnegative integers $\hat{m}$ and $s$, we set

$$m_k = \begin{cases} 
0 & \text{if } k \leq s \\
\min \{m_{k-1} + 1, \hat{m}\} & \text{otherwise} 
\end{cases}.$$  \hfill (20)

Throughout the experiments, unless otherwise stated, we used the following parameters:

$$\hat{m} = 5, \quad s = 5, \quad \beta = 0.3, \quad \text{and} \quad \sigma = 1.0e - 4.$$ \hfill (21)

For the Fischer-Burmeister merit function (FBMF, for short) approach [4], we chose a limited-memory BFGS algorithm with 5 limited-memory vector-updates [3] to solve the unconstrained minimization reformulation (8) for the SOCCP (1). For the scaling matrix $H^0 = \gamma I$ in the BFGS algorithm, we adopted the choice of $\gamma = p^T q / p^T q$ recommended by [27, P. 226], where $p := \zeta - \zeta^{\text{old}}$ and $q := \nabla f_{\text{FB}}(\zeta) - \nabla f_{\text{FB}}(\zeta^{\text{old}})$. To ensure convergence, we revert to the steepest descent direction $-\nabla f_{\text{FB}}(\zeta)$ whenever the current direction $\Delta$ fails to satisfy the sufficient descent condition

$$\nabla f_{\text{FB}}(\zeta)^T \Delta \leq -10^{-4} \|\nabla f_{\text{FB}}(\zeta)\| \|\Delta\|.$$

In addition, we also employed the same nonmonotone line search as above to seek a suitable step-length, except that the parameter $\beta$ is chosen as 0.2.

During the experiments, we started Algorithm 3.1 and the FBMF approach with the starting point $\zeta^0 = 0.001(1,1,\cdots,1)^T$ and terminated the iterate once one of the following conditions is satisfied:

1. \( \max \{\Psi(\zeta), |F(\zeta)^T \zeta|\} \leq 10^{-4} \), where $\Psi$ represents $f_{\text{VF}}$ or $f_{\text{FB}}$.

2. The number of iteration is over 50000.

3. The step-length is lower than $10^{-16}$.

We have done the following three groups of experiments.

**Experiment A.** Testing the influence of the scaling strategy on Algorithm 3.1 and the FBMF method. Note that, when the matrix $M$ and the vector $q$ in (18) are replaced by

$$\tilde{M} = M / w \quad \text{and} \quad \tilde{q} = q / w$$ \hfill (22)

where $w \geq 1$ is a constant, the optimal solution of problem (18) does not change. Hence, in this experiment, we generated 10 test problems with sparsity 0.5% and 10% and $m = 10$, $n_1 = n_2 = \cdots = n_m = 100$, and then solved each problem and their different scaled formulations with Algorithm 3.1 and the FBMF approach. Numerical results are summarized in Tables 1 and 2, where $N_0$, represents the number of problem, $\text{Den}$ denotes the approximate sparsity of $M$, $\text{Nf}$ and $\text{Time}$ respectively denote the total number of function evaluations and the CPU time for solving each problem.
Table 1: Numerical results of Algorithm 3.1 for the scaled problems

<p>| | | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>N0</td>
<td>Den</td>
<td>Nf</td>
<td>Time</td>
<td>Nf</td>
<td>Time</td>
<td>Nf</td>
<td>Time</td>
<td>Nf</td>
<td>Time</td>
</tr>
<tr>
<td>1</td>
<td>0.5%</td>
<td>21963</td>
<td>69.68</td>
<td>6716</td>
<td>21.71</td>
<td>4201</td>
<td>15.31</td>
<td>5608</td>
<td>21.89</td>
</tr>
<tr>
<td>2</td>
<td>0.5%</td>
<td>55916</td>
<td>175.9</td>
<td>26234</td>
<td>85.81</td>
<td>17836</td>
<td>68.15</td>
<td>24073</td>
<td>98.15</td>
</tr>
<tr>
<td>3</td>
<td>0.5%</td>
<td>11897</td>
<td>44.57</td>
<td>989</td>
<td>3.82</td>
<td>803</td>
<td>3.40</td>
<td>1168</td>
<td>5.34</td>
</tr>
<tr>
<td>4</td>
<td>0.5%</td>
<td>14860</td>
<td>53.68</td>
<td>998</td>
<td>4.04</td>
<td>776</td>
<td>3.28</td>
<td>1047</td>
<td>5.07</td>
</tr>
<tr>
<td>5</td>
<td>0.5%</td>
<td>13260</td>
<td>48.53</td>
<td>553</td>
<td>2.01</td>
<td>553</td>
<td>2.32</td>
<td>733</td>
<td>3.20</td>
</tr>
<tr>
<td>6</td>
<td>10%</td>
<td>–</td>
<td>–</td>
<td>2238</td>
<td>89.67</td>
<td>237</td>
<td>10.09</td>
<td>99</td>
<td>4.46</td>
</tr>
<tr>
<td>7</td>
<td>10%</td>
<td>–</td>
<td>–</td>
<td>2518</td>
<td>95.54</td>
<td>264</td>
<td>10.64</td>
<td>114</td>
<td>5.21</td>
</tr>
<tr>
<td>8</td>
<td>10%</td>
<td>–</td>
<td>–</td>
<td>8592</td>
<td>344.4</td>
<td>228</td>
<td>10.26</td>
<td>162</td>
<td>7.23</td>
</tr>
<tr>
<td>9</td>
<td>10%</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>273</td>
<td>12.78</td>
<td>81</td>
<td>6.18</td>
</tr>
<tr>
<td>10</td>
<td>10%</td>
<td>–</td>
<td>–</td>
<td>1982</td>
<td>82.60</td>
<td>239</td>
<td>10.98</td>
<td>125</td>
<td>5.56</td>
</tr>
</tbody>
</table>

where “−” means that the iteration was stopped since the step-length was less than $10^{-16}$.

From Tables 1 and 2, we see that, when $w > 1$, i.e. imposing the scaling strategy on the original problems, Algorithm 3.1 and the FBMF approach require much less function evaluations. Therefore, the scaling strategy in (22) can greatly improve the numerical performance of Algorithm 3.1 and the merit function approach. In particular, for those problems to which Algorithm 3.1 fails due to too small step-length, using the scaling strategy can yield satisfying solutions. This implies that Algorithm 3.1 has more dependence on the scaling strategy than the MF approach.

**Experiment B.** Testing Algorithm 3.1 and the FBMF approach on the affine SOCCP (18) with various degree of sparsity. In this experiment, we generated 10 test problems with $m = 1$ and $n = 1000$ for each nonzero density 0.1%, 0.5%, 1%, 10%, 50% and 80%, and then solved each problem with Algorithm 3.1 and the FBMF approach. Numerical results were summarized in Tables 3-4, where **Nf** and **Time** are same as Experiment A, **Gap** denotes the value of $|F(\zeta)^T\zeta|$ at the final iteration, and **Scale** in Table 4 denotes the value of $w$ in (22). In particular, the values of **Gap**, **Nf** and **Time** in Table 4 are the averages of 10 trials for each sparsity.

From Table 3, it appears that Algorithm 3.1 and the FBMF approach have similar nu-
Table 2: Numerical results of the FBMF method for the scaled problems

<table>
<thead>
<tr>
<th>N0.</th>
<th>Den</th>
<th>( w = 1 )</th>
<th>( w = 10 )</th>
<th>( w = 50 )</th>
<th>( w = 100 )</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Nf</td>
<td>Time</td>
<td>Nf</td>
<td>Time</td>
</tr>
<tr>
<td>1</td>
<td>0.5%</td>
<td>8135</td>
<td>56.96</td>
<td>4346</td>
<td>30.01</td>
</tr>
<tr>
<td>2</td>
<td>0.5%</td>
<td>9086</td>
<td>57.56</td>
<td>14020</td>
<td>91.06</td>
</tr>
<tr>
<td>3</td>
<td>0.5%</td>
<td>611</td>
<td>3.95</td>
<td>531</td>
<td>3.70</td>
</tr>
<tr>
<td>4</td>
<td>0.5%</td>
<td>1030</td>
<td>7.65</td>
<td>677</td>
<td>5.17</td>
</tr>
<tr>
<td>5</td>
<td>0.5%</td>
<td>769</td>
<td>5.56</td>
<td>403</td>
<td>2.98</td>
</tr>
<tr>
<td>6</td>
<td>10%</td>
<td>6682</td>
<td>488.0</td>
<td>807</td>
<td>64.15</td>
</tr>
<tr>
<td>7</td>
<td>10%</td>
<td>4668</td>
<td>337.7</td>
<td>737</td>
<td>56.85</td>
</tr>
<tr>
<td>8</td>
<td>10%</td>
<td>5639</td>
<td>431.1</td>
<td>812</td>
<td>63.82</td>
</tr>
<tr>
<td>9</td>
<td>10%</td>
<td>4616</td>
<td>347.4</td>
<td>723</td>
<td>57.21</td>
</tr>
<tr>
<td>10</td>
<td>10%</td>
<td>5818</td>
<td>452.6</td>
<td>702</td>
<td>59.12</td>
</tr>
</tbody>
</table>

Numerical performance on those problems with sparsity 0.1%. However, from Table 4, we see that, under the scaling strategy shown, Algorithm 3.1 always needed less CPU time than the FBMF approach although the former may require more function evaluations. In addition, we also observe that the number of function evaluations required by Algorithm 3.1 will become less when the sparsity of \( M \) becomes higher.

**Experiment C.** Testing Algorithm 3.1 and the FBMF approach on the affine SOCCP (18) with various Cartesian structures of \( \mathcal{K} \). To construct SOCs of various types, we chose \( n_i \) and \( m \) such that \( n_1 = n_2 = \cdots = n_m \) and \( n_1 + \cdots + n_m = 2000 \). For each type of \( \mathcal{K} \), we solved 10 test problems with nonzero density 1% by Algorithm 3.1 and the FBMF approach, respectively. Numerical results were reported in Table 5, where **Scale**, **Gap**, **Nf** and **Time** are same as Experiment A, and particularly the values of **Gap**, **Nf** and **Time** are the averages of 10 trials for each type of \( \mathcal{K} \).

From Table 5, we see that, under the scaling strategy shown, Algorithm 3.1 is comparable with the FBMF method for the first five groups of test problems whether in the CPU time or in the number of function evaluations. For the last group of test problems, Algorithm 3.1 obviously required more CPU time and function evaluations than the FBMF approach. However, from Table 6, we see that if **Scale** is still chosen as 100 but
Table 3: Numerical results for the affine SOCCP with sparsity 0.1%

<table>
<thead>
<tr>
<th>N0.</th>
<th>Nf</th>
<th>Time</th>
<th>N0.</th>
<th>Nf</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>597</td>
<td>0.76</td>
<td>2</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>3</td>
<td>539</td>
<td>0.85</td>
<td>4</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>5</td>
<td>*</td>
<td>*</td>
<td>6</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>7</td>
<td>254</td>
<td>0.34</td>
<td>8</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>9</td>
<td>*</td>
<td>*</td>
<td>10</td>
<td>799</td>
<td>0.95</td>
</tr>
</tbody>
</table>

where “*” means that the iteration was stopped since the number of iteration was over 50000.

Table 4: Numerical results for the affine SOCCP with different sparsity

<table>
<thead>
<tr>
<th>Den</th>
<th>Scale</th>
<th>Algorithm 3.1</th>
<th>MF approach</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Gap</td>
<td>Nf</td>
</tr>
<tr>
<td>0.5%</td>
<td>1</td>
<td>9.16e-5</td>
<td>1597.1</td>
</tr>
<tr>
<td>1%</td>
<td>1</td>
<td>8.10e-5</td>
<td>7401.4</td>
</tr>
<tr>
<td>10%</td>
<td>10</td>
<td>6.89e-5</td>
<td>402.6</td>
</tr>
<tr>
<td>50%</td>
<td>100</td>
<td>5.90e-5</td>
<td>472.2</td>
</tr>
<tr>
<td>80%</td>
<td>100</td>
<td>4.81e-5</td>
<td>468.4</td>
</tr>
</tbody>
</table>

the parameter \( \beta \) in the line search is chosen as 0.1 instead of 0.3, the numerical performance of Algorithm 3.1 will have a great improvement, and moreover, the CPU time and the number of function evaluations needed are comparable with those of the FBMF method.

To sum up, for the symmetric affine SOCCPs in (18), if a suitable scaling strategy and the parameter \( \beta \) are used, Algorithm 3.1 will be comparable with, even superior to, the FBMF method in the CPU time for solving test problems although the former may
require more function evaluations. Otherwise, the FBMF approach will be superior to Algorithm 3.1 whether in the CPU time or in the number of function evaluations.

Table 5: Numerical results for the affine SOCCP with different $K$

<table>
<thead>
<tr>
<th>$m$</th>
<th>Scale</th>
<th>Algorithm 3.1</th>
<th>MF approach</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Gap</td>
<td>Nf</td>
</tr>
<tr>
<td>1</td>
<td>10</td>
<td>7.95e-5</td>
<td>201.3</td>
</tr>
<tr>
<td>10</td>
<td>10</td>
<td>8.24e-5</td>
<td>497.6</td>
</tr>
<tr>
<td>50</td>
<td>10</td>
<td>9.34e-5</td>
<td>1193</td>
</tr>
<tr>
<td>100</td>
<td>100</td>
<td>5.75e-5</td>
<td>116.4</td>
</tr>
<tr>
<td>200</td>
<td>100</td>
<td>4.54e-5</td>
<td>129.2</td>
</tr>
<tr>
<td>500</td>
<td>100</td>
<td>7.09e-5</td>
<td>9719.4</td>
</tr>
</tbody>
</table>

Table 6: Numerical results of Algorithm 3.1 for different $\beta$

<table>
<thead>
<tr>
<th>$\beta = 0.3$</th>
<th>$\beta = 0.1$</th>
<th>$\beta = 0.3$</th>
<th>$\beta = 0.1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>N0.</td>
<td>Nf</td>
<td>Time</td>
<td>N0.</td>
</tr>
<tr>
<td>1</td>
<td>75</td>
<td>8.42</td>
<td>198</td>
</tr>
<tr>
<td>3</td>
<td>19001</td>
<td>2231.4</td>
<td>210</td>
</tr>
<tr>
<td>5</td>
<td>69</td>
<td>7.93</td>
<td>178</td>
</tr>
<tr>
<td>7</td>
<td>62169</td>
<td>7068.4</td>
<td>132</td>
</tr>
<tr>
<td>9</td>
<td>77</td>
<td>8.54</td>
<td>208</td>
</tr>
</tbody>
</table>

5 Final Remarks

In this paper, we investigated a descent method for the equivalent reformulation (11) of the SOCCP which was also used for the NCP and the SDCP in literature, and proved
its global convergence under some mild assumptions. Numerical comparison with the Fischer-Burmeister merit function approach [4] for symmetric affine SOCCPs generated randomly indicate that the descent method is comparable with, even to superior to, the FBMF approach in the CPU time if a suitable scaling strategy and the parameter $\beta$ in line search are adopted. We also expect that the method can be used to deal with large SOCCPs due to very small computational work per iteration. In addition, we notice that the proposed algorithm does not work for another reformulation (8) of the SOCP since $f_{\text{FB}}$ lacks property of bounded level sets (Lem. 2.3) where $\psi_0$ plays an important role therein.

Prop. 3.1-3.2 are more or less an afterthought of [4], nonetheless, it does parallel the extension to the SOCP from the NCP and SDCP cases. On the other hand, this work does a further study based on replacing the conditions of monotonicity and strict feasibility by a new (and weaker under certain sense) so-called $R_{01}$-function. More specifically, under the new so-called $R_{01}$-function condition, the level sets of $f_{\text{YF}}$ are still bounded and the proposed descent algorithm still has global convergence. These results are significant not only they are brand-new but also there is no needs the assumption of strict feasibility therein.

One future topic is to analyze the convergence rate theoretically which is more intractable. Other direction like weakening conditions which guarantees the property of bounded level sets is also interesting and worthwhile. There may have the direction as one referee pointed out which is to apply this optimization method to real-life studies, for example [5] and references therein.

Acknowledgement. The authors thank the referees for their careful reading of the paper and helpful suggestions.

References


mulation of the second-order cone complementarity problem, Mathematical Program-


[6] X.-D. Chen, D. Sun, and J. Sun(2003), Complementarity functions and numerical
experiments for second-order cone complementarity problems, Computational Optim-


[8] F. Facchinei and J. Soares(1997), A new merit function for nonlinear comple-


[10] M. C. Ferris and J.-S. Pang(1997), Engineering and economic applications of


Interscience, Chichester.


