A Matrix Splitting Method for Symmetric Affine Second-Order Cone Complementarity Problems*

Shunsuke Hayashi, Takahiro Yamaguchi, Nobuo Yamashita and Masao Fukushima†

June 30, 2003

Abstract

The affine Second-Order Cone Complementarity Problem (SOCCP) is a wide class of problems that contains the Linear Complementarity Problem (LCP) as a special case. The purpose of this paper is to propose an iterative method for the symmetric affine SOCCP that is based on the idea of matrix splitting. Matrix splitting methods have originally been developed for the solution of the system of linear equations and have subsequently been extended to the linear complementarity problem (LCP) and the affine variational inequality problem. In this paper, we first give conditions under which the matrix splitting method converges to a solution of the affine SOCCP. We then present, as a particular realization of the matrix splitting method, the block successive overrelaxation (SOR) method for the affine SOCCP involving a positive definite matrix, and propose an efficient method for solving subproblems. Finally, we report some numerical results with the proposed algorithm, where promising results are obtained especially for problems with sparse matrices.

Keywords: Second-order cone complementarity problem; Matrix splitting method; Successive overrelaxation method

1 Introduction

The l-dimensional *second order cone* (SOC) is defined by

$$\mathcal{K}^l = \{(x_1, x_2) \in \Re \times \Re^{l-1} \mid x_1 \geq \|x_2\|\},$$

where $\|\cdot\|$ denotes the Euclidean norm. In particular, if $l = 1$, $\mathcal{K}^1$ is the set of nonnegative reals.

The complementarity problem over SOCs is called the *second-order cone complementarity problem* (SOCCP) [9]. The SOCCP contains the nonlinear complementarity problem (NCP) as a special case. Moreover, the Karush-Kuhn-Tucker (KKT) conditions of the second-order cone programming problem (SOCP) [1, 17] can be written in the form of SOCCP.

In this paper, we focus on the symmetric affine SOCCP :

$$\begin{array}{rl} \text{Find} & z \in \Re^n \\ \text{such that} & z \in \mathcal{K},\ \ Mz + q \in \mathcal{K},\ \ z^T(Mz+q) = 0, \end{array} \tag{1.1}$$

where $M \in \Re^{n\times n}$ and $q \in \Re^n$ are a given symmetric matrix and a vector, respectively, and $\mathcal{K} \subset \Re^n$ is the Cartesian product of SOCs, i.e.,

$$\mathcal{K} = \mathcal{K}^{n_1} \times \mathcal{K}^{n_2} \times \cdots \times \mathcal{K}^{n_m}$$

*This research was supported in part by a Grant-in-Aid for Scientific Research (B) from the Ministry of Education, Science, Sports and Culture of Japan.

†Department of Applied Mathematics and Physics, Graduate School of Informatics, Kyoto University, Kyoto 606-8501, Japan ({hayashi, yamagchi}@amp.i.kyoto-u.ac.jp, {nobuo, fuku}@i.kyoto-u.ac.jp)

with $n = n_1 + n_2 + \cdots + n_m$. For instance, the KKT conditions for the Lagrangian dual problem of the SOCP :

$$\begin{array}{ll} \text{Minimize} & \frac{1}{2} z^T Q z + c^T z \\ \text{subject to} & Az + b \in \mathcal{K} \end{array}$$

can be written in the form of symmetric affine SOCCP (1.1), where Q is a symmetric positive definite matrix. Throughout the paper, we often denote SOCCP (1.1) as SOCCP $(q, M, \mathcal{K})$.

Recently, several methods have been proposed for solving SOCCPs. One of the popular approaches is to reformulate the SOCCP into an equivalent nondifferentiable minimization problem and solve it by smoothing methods [3, 5, 9, 13]. Such an approach is motivated by smoothing methods for nonlinear complementarity problems [2, 4, 16, 21]. However, smoothing methods may sometimes be expensive computationally for large scale problems, and hence, methods exploiting particular features of matrices such as the sparsity and the block structure are required. In this paper, we propose an approach based on a matrix splitting method for the affine SOCCP.

In matrix splitting methods, we represent the matrix M as the sum of two matrices B and C where B has a certain simple structure. Then, by solving subproblems involving the matrix B successively, the method generates a sequence converging to a solution of the original problem. Historically, matrix splitting methods have been used to solve the system of linear equations [11]. Several splitting schemes such as Jacobi, Gauss-Seidel, and the successive overrelaxation (SOR) methods have been suggested and various parallel algorithms have been developed on the basis of those matrix splitting schemes. Subsequently, those methods have been extended to the linear complementarity problem (LCP) and the affine variational inequality problem [7, 15, 18, 19, 20, 22].

In this paper, we extend the matrix splitting method [7] for LCP to SOCCP. Especially, we propose to adopt the the block SOR method. The block SOR method for LCP generates subproblems that can be solved easily by simple substitution operations. However, if the method is applied to SOCCP in a direct manner, generated subproblems need not be tractable in general. We will present a special block SOR method for SOCCP in which generated subproblems possess certain particular structures. In our approach, the subproblems are transformed into equivalent single variable equations by exploiting the special structures of subproblems, and then, those equations are solved efficiently by adopting some idea used in the trust region method.

The paper is organized as follows: In Section 2, we describe the basic matrix splitting method for affine SOCCPs, and give conditions for the method to be convergent. In Section 3, we present the block SOR method as a particular realization of the basic splitting method of Section 2. In Section 4, we give a concrete procedure to solve subproblems of the block SOR method. In Section 5, we report numerical results with the proposed method. Finally, we conclude the paper in Section 6.

2 Matrix splitting method and its convergence

In this section, we extend the matrix splitting method for LCP to SOCCP, and study its convergence property. Let the symmetric matrix M be represented as the sum of two matrices $B \in \Re^{n \times n}$ and $C \in \Re^{n \times n}$, i.e.,

$$M = B + C, \tag{2.1}$$

where B and C need not be symmetric. Such a pair (B, C) is called a splitting of M. The basic algorithm of the matrix splitting method for the SOCCP is stated as follows:

Algorithm 2.1

Step 1. *Choose a splitting (B, C) of M and an initial point $z^0 \in \mathcal{K}$. Set $k := 0$.*

Step 2. *Solve the following affine SOCCP*:

$$\begin{aligned} &\textit{Find} \quad z \in \Re^n \\ &\textit{such that} \quad z \in \mathcal{K},\ Bz + q^k \in \mathcal{K},\ z^T(Bz + q^k) = 0, \end{aligned} \tag{2.2}$$

where

$$q^k := q + Cz^k. \tag{2.3}$$

Let z^{k+1} be a solution of problem (2.2).

Step 3. *If $z^{k+1} = z^k$, terminate. Otherwise, return to Step* 2 *with k replaced by $k + 1$.*

It is particularly important to choose a splitting (B, C) so that SOCCP (2.2) can be solved efficiently and the sequence $\{z^k\}$ generated by the algorithm converges to a solution of SOCCP (1.1).

Next, by extending the convergence theory of splitting methods for LCP [7, 15] to SOCCP, we give conditions for Algorithm 2.1 to be convergent. The following two definitions are necessary for describing the conditions.

Definition 2.1 *If SOCCP$(q, B, \mathcal{K})$ has a solution for any $q \in \Re^n$, then B is called a $\mathcal{K}$-Q-matrix. Moreover, if B is a $\mathcal{K}$-Q-matrix, then (B, C) is called a $\mathcal{K}$-Q-splitting.*

Definition 2.2 *If $B - C$ is positive (semi-)definite, then the splitting (B, C) is said to be (weakly) regular.*

It is evident from the definition that SOCCP (2.2) always has a solution if (B, C) is a $\mathcal{K}$-Q-splitting. On the other hand, the regularity of the splitting (B, C) plays an important role in discussing the convergence property of Algorithm 2.1.

In what follows, we will show that, if (B, C) is a regular $\mathcal{K}$-Q-splitting, then the sequence generated by Algorithm 2.1 converges to a solution of SOCCP (1.1). To this end, we introduce function $\theta : \Re^n \to \Re$ defined by

$$\theta(z) := \frac{1}{2} z^T M z + q^T z.$$

This function serves as a merit function on $\mathcal{K}$ for SOCCP (1.1) since matrix M is symmetric and hence a minimizer of θ on $\mathcal{K}$ solves SOCCP (1.1). In the subsequent discussions, we will use function θ in order to monitor the progress of the algorithm. The following lemma gives a sufficient condition for the sequence $\{\theta(z^k)\}$ to be nonincreasing.

Lemma 2.1 *Let (B, C) be a weakly regular $\mathcal{K}$-Q-splitting of the symmetric matrix M. Let $\{z^k\}$ be a sequence generated by Algorithm* 2.1*. Then, we have for each k*

$$\theta(z^k) - \theta(z^{k+1}) \geq \frac{1}{2}(z^k - z^{k+1})^T(B - C)(z^k - z^{k+1}) \geq 0. \tag{2.4}$$

In particular, if (B, C) is a regular $\mathcal{K}$-Q-splitting, then the second inequality in (2.4) *holds strictly whenever $z^k \neq z^{k+1}$, and moreover $\theta(z^k) = \theta(z^{k+1})$ if and only if $z^k = z^{k+1}$.*

Proof. By (2.1), (2.3) and the symmetry of M, we have

$$\begin{aligned} \theta(z^k) - \theta(z^{k+1}) &= (z^k - z^{k+1})^T(q + Mz^{k+1}) + \frac{1}{2}(z^k - z^{k+1})^T M (z^k - z^{k+1}) \\ &= (z^k - z^{k+1})^T(q + (B + C)z^{k+1}) \\ &\quad + \frac{1}{2}(z^k - z^{k+1})^T(B - C)(z^k - z^{k+1}) + (z^k - z^{k+1})^T C (z^k - z^{k+1}) \\ &= (z^k - z^{k+1})^T(q^k + Bz^{k+1}) + \frac{1}{2}(z^k - z^{k+1})^T(B - C)(z^k - z^{k+1}). \end{aligned}$$

Moreover, $(z^k)^T(q^k + Bz^{k+1}) \ge 0$ since $z^k \in \mathcal{K}$ and $q^k + Bz^{k+1} \in \mathcal{K}$, and $(z^{k+1})^T(q^k + Bz^{k+1}) = 0$ since z^{k+1} is a solution of SOCCP (2.2). We then have $(z^k - z^{k+1})^T(q^k + Bz^{k+1}) \ge 0$ and hence

$$\theta(z^k) - \theta(z^{k+1}) \ge \frac{1}{2}(z^k - z^{k+1})^T(B - C)(z^k - z^{k+1}).$$

The second inequality in (2.4) holds from the positive semidefiniteness of $B - C$. The last assertion of the lemma is obtained by the fact that the regularity of (B, C) implies the positive definiteness of $B - C$. ∎

The above lemma leads to the following theorem.

Theorem 2.1 *Let (B, C) be a regular $\mathcal{K}$-Q-splitting of the symmetric matrix M. Then, any accumulation point of the sequence $\{z^k\}$ generated by Algorithm 2.1 is a solution of SOCCP (1.1).*

Proof. Let $\tilde{z}$ be an arbitrary accumulation point of the sequence $\{z^k\}$ generated by Algorithm 2.1 and $\{z^{k_i}\}$ be a subsequence of $\{z^k\}$ converging to $\tilde{z}$. Then, by the continuity of θ, the sequence $\{\theta(z^{k_i})\}$ converges to $\theta(\tilde{z})$. In addition, the entire sequence $\{\theta(z^k)\}$ is bounded below, since $\{\theta(z^k)\}$ is nonincreasing from Lemma 2.1 and the subsequence $\{\theta(z^{k_i})\}$ converges. Consequently, the sequence $\{\theta(z^k)\}$ itself converges. This fact, along with the positive definiteness of $B - C$ and the inequality (2.4), yields that $\{z^k - z^{k+1}\}$ converges to 0. Hence, the sequence $\{z^{k_i+1}\}$ also converges to $\tilde{z}$. Since z^{k_i+1} satisfies

$$\begin{aligned} z^{k_i+1} &\in \mathcal{K} \\ Bz^{k_i+1} + Cz^{k_i} + q &\in \mathcal{K} \\ (z^{k_i+1})^T(Bz^{k_i+1} + Cz^{k_i} + q) &= 0, \end{aligned}$$

passing to the limit reveals that $\tilde{z}$ is a solution of SOCCP (1.1). ∎

This theorem indicates that, if a sequence $\{z^k\}$ generated by Algorithm 2.1 has an accumulation point, then it is a solution of SOCCP (1.1). However, the theorem says nothing about the existence of an accumulation point. In order to show the boundedness of $\{z^k\}$, we introduce the following concept on M.

Definition 2.3 *A matrix $M \in \Re^{n\times n}$ is said to be (strictly) $\mathcal{K}$-copositive if $z^T Mz \ge (>)\, 0$ for all $z \in \mathcal{K}\backslash\{0\}$.*

The concept of (strict) $\mathcal{K}$-copositivity is a natural extension of the (strict) copositivity in the LCP theory [7]. It is easily seen that any positive semidefinite matrix is $\mathcal{K}$-copositive and any positive definite matrix is strictly $\mathcal{K}$-copositive. Using this definition, we establish the main theorem in this section.

Theorem 2.2 *Let M be a symmetric matrix, and q be an arbitrary vector. If M is strictly $\mathcal{K}$-copositive, then, for any initial point $z^0 \in \mathcal{K}$, the sequence $\{z^k\}$ generated by Algorithm 2.1 with regular $\mathcal{K}$-Q-splitting (B, C) is bounded, and its arbitrary accumulation point is a solution of SOCCP (1.1).*

Proof. From Theorem 2.1, if the generated sequence has an accumulation point, it is a solution of SOCCP (1.1). Hence, it suffices to show the boundedness of the sequence $\{z^k\}$.

Since M is strictly $\mathcal{K}$-copositive, we have

$$\sigma := \min_{\substack{\|e\|=1 \\ e\in\mathcal{K}}} e^T Me > 0,$$

that is, we have for any $z \in \mathcal{K}$

$$z^T Mz \ge \sigma\|z\|^2. \tag{2.5}$$

Therefore, we have for all k

$$\begin{aligned}\theta(z^0) &\geq \theta(z^k)\\ &= \frac{1}{2}(z^k)^T M z^k + q^T z^k\\ &\geq \frac{1}{2}\sigma\|z^k\|^2 - \|q\|\|z^k\|\\ &= \frac{1}{2}\sigma\Big(\|z^k\| - \frac{1}{\sigma}\|q\|\Big)^2 - \frac{1}{2\sigma}\|q\|^2,\end{aligned}$$

where the first inequality follows from Lemma 2.1, the second inequality follows from (2.5), $\{z^k\} \subset \mathcal{K}$ and the Cauchy-Schwarz inequality. This inequality readily yields

$$\|z^k\| \leq \frac{1}{\sigma}\|q\| + \sqrt{\frac{2}{\sigma}\left(\theta(z^0) + \frac{1}{2\sigma}\|q\|^2\right)}$$

for all k, which shows the boundedness of $\{z^k\}$. ■

3 Block SOR method

In the previous section, we have shown that, under the assumption that M is strictly $\mathcal{K}$-copositive, a sequence generated by Algorithm 2.1 with any regular $\mathcal{K}$-Q-splitting (B, C) converges to a solution of SOCCP (1.1). In this section, we present the block successive overrelaxation (block SOR) method for solving affine SOCCPs by extending the corresponding method for LCPs [7]. In particular, we give conditions for a splitting (B, C) used in the block SOR method to be a regular $\mathcal{K}$-Q-splitting. In this and the next sections, we suppose that the following assumption holds.

Assumption A *The symmetric matrix $M \in \Re^{n\times n}$ is positive definite.*

Note that the matrix M is strictly $\mathcal{K}$-copositive under this assumption.

First, we give an explicit expression of the splitting (B, C). Let the matrix M be partitioned as

$$M = \begin{pmatrix} M_{11} & M_{12} & \cdots & M_{1m}\\ M_{21} & M_{22} & & M_{2m}\\ \vdots & & \ddots & \vdots\\ M_{m1} & M_{m2} & \cdots & M_{mm}\end{pmatrix}$$

with $M_{ij} \in \Re^{n_i\times n_j}$. Then, the splitting (B, C) used in the block SOR method is represented as

$$B = \begin{pmatrix} B_{11} & & & & \text{\Large 0}\\ M_{21} & B_{22} & & & \\ M_{31} & M_{32} & \ddots & & \\ \vdots & & \ddots & \ddots & \\ M_{m1} & \dots & \dots & M_{m,m-1} & B_{mm}\end{pmatrix}, \quad C = M - B, \tag{3.1}$$

where B_{ii} are chosen appropriately for $i = 1, \dots, m$. Since B is chosen to be a block lower triangular matrix, we can solve SOCCP (2.2) successively as follows: Let z and q^k in SOCCP (2.2) be partitioned as

$$z = \begin{pmatrix} z_1\\ \vdots\\ z_m\end{pmatrix}, \quad q^k = \begin{pmatrix} q_1^k\\ \vdots\\ q_m^k\end{pmatrix},$$

where $z_i \in \Re^{n_i}$ and $q_i^k \in \Re^{n_i}$, $i = 1, \ldots, m$. Then the decomposable structure of SOC constraints [13, Proposition 2.2] yields that SOCCP (2.2) is equivalent to the problem of finding $z \in \Re^n$ such that

$$z_i \in \mathcal{K}^{n_i}, \quad B_{ii} z_i + r_i^k \in \mathcal{K}^{n_i}, \quad z_i^T (B_{ii} z_i + r_i^k) = 0 \quad i = 1, \ldots, m, \tag{3.2}$$

where

$$r_i^k := \begin{cases} q_1^k & \text{if } i = 1, \\ \displaystyle\sum_{j=1}^{i-1} M_{ij} z_j + q_i^k & \text{if } i \geq 2 \ . \end{cases}$$

We can solve problems (3.2) for z_i recursively from $i = 1$ to $i = m$, by regarding $z_1, \ldots, z_{i-1}$ and r_i^k as known constants.

In the block SOR method for LCP, the block diagonal elements B_{ii} are normally chosen as $B_{ii} := w^{-1} M_{ii}$ with a constant $\omega \in (0, 2)$. However, in the case of SOCCP, subproblems (3.2) may not be solved efficiently if $B_{ii} = w^{-1} M_{ii}$. Here, we propose to choose B_{ii} as follows: To simplify the notation, we introduce the function $\Gamma : \Re^{l \times l} \times (0, +\infty) \times [0, +\infty) \to \Re^{l \times l}$ defined by

$$\Gamma(A, \omega, \gamma) := \begin{cases} \omega^{-1} a_1 & (l = 1) \\ \begin{pmatrix} \omega^{-1} a_1 & 0^T \\ \gamma a_2 & \omega^{-1} A_3 \end{pmatrix} & (l \geq 2) \,, \end{cases} \tag{3.3}$$

where $\omega > 0$, $\gamma \geq 0$, and $A \in \Re^{l \times l}$ is given by

$$A = \begin{pmatrix} a_1 & a_2^T \\ a_2 & A_3 \end{pmatrix} \tag{3.4}$$

with $a_1 \in \Re$, $a_2 \in \Re^{l-1}$ and $A_3 \in \Re^{(l-1) \times (l-1)}$. Using this function, we let

$$B_{ii} := \Gamma(M_{ii}, \omega, \gamma).$$

SOCCP (3.2) can be solved efficiently by exploiting the particular structure of $B_{ii} = \Gamma(M_{ii}, \omega, \gamma)$, as will be shown in the next section.

Next we consider conditions for the splitting (3.1) with $B_{ii} = \Gamma(M_{ii}, \omega, \gamma)$ to be a regular $\mathcal{K}$-Q-splitting. To this end, we give three lemmas and a proposition in the following.

Lemma 3.1 [14, Theorem 7.7.6] *Let $A \in \Re^{l \times l}$ be a symmetric matrix given by* (3.4)*. Then, A is positive definite if and only if $a_1 > 0$ and $A_3 - a_1^{-1} a_2 a_2^T$ (the Schur complement of A with respect to a_1) is positive definite.*

Lemma 3.2 *Let $A \in \Re^{l \times l}$ be given by* (3.4) *and $G \in \Re^{l \times l}$ be given by $G = \Gamma(A, \omega, \gamma)$. If* (i) *$\gamma = 0$ and $\omega > 0$ or* (ii) *$\gamma > 0$ and $0 < \omega \leq 2/\gamma$, then the matrix G is positive definite.*

Proof. Since the lemma holds evidently for $l = 1$, we only consider the case $l \geq 2$. In case (i), we can easily obtain the positive definiteness of $G = \Gamma(A, \omega, 0)$ from $\omega > 0$ and the positive definiteness of A. Let us consider case (ii). We will show the positive definiteness of $(G + G^T)/2$ instead of G. From

$$(G + G^T)/2 = \begin{pmatrix} \omega^{-1} a_1 & (\gamma/2) a_2^T \\ (\gamma/2) a_2 & \omega^{-1} A_3 \end{pmatrix}$$

and Lemma 3.1, it is sufficient to show that $\omega^{-1}a_1 > 0$ and that the Schur complement of $(G+G^T)/2$ with respect to $\omega^{-1}a_1$ is positive definite. From $\omega > 0$ and the positive definiteness of A, we readily obtain $\omega^{-1}a_1 > 0$. The Schur complement of $(G+G^T)/2$ with respect to $\omega^{-1}a_1$, which is written as

$$\omega^{-1}A_3 - \frac{(\gamma^2/4)a_2a_2^T}{\omega^{-1}a_1} = \frac{\omega\gamma^2}{4}\left\{\left(\frac{4}{\omega^2\gamma^2}-1\right)A_3 + \left(A_3 - \frac{a_2a_2^T}{a_1}\right)\right\}, \tag{3.5}$$

is positive definite by the following reason: Since matrix A is positive definite, A_3 and $A_3 - a_1^{-1}a_2a_2^T$ are positive definite from Lemma 3.1. In addition, it follows from $0 < \omega \leq 2/\gamma$ that $\omega\gamma^2/4 > 0$ and $4/(\omega^2\gamma^2) - 1 \geq 0$. Hence, the matrix given by (3.5) is positive definite. This completes the proof. ■

Lemma 3.3 *Let $A \in \Re^{l\times l}$ be a positive definite symmetric matrix given by* (3.4) *and (G, H) be a splitting of A given by*

$$G = \Gamma(A,\omega,\gamma), \quad H = A - \Gamma(A,\omega,\gamma).$$

Suppose that ω and γ satisfy either of the following conditions:

$$\begin{aligned} &\text{(a)} \quad \gamma > 1 \ \textit{and} \ 0 < \omega \leq 2/\gamma, \\ &\text{(b)} \quad \gamma = 1 \ \textit{and} \ 0 < \omega < 2, \\ &\text{(c)} \quad 0 \leq \gamma < 1 \ \textit{and} \ 0 < \omega \leq 2/(2-\gamma). \end{aligned}$$

Then the splitting (G, H) is regular.

Proof. Since the lemma holds evidently for $l = 1$, we only consider the case $l \geq 2$. We will show the positive definiteness of $((G-H)+(G-H)^T)/2$ instead of $G - H$. We have from

$$\begin{aligned} G - H &= 2\Gamma(A,\omega,\gamma) - A \\ &= \begin{pmatrix} (2\omega^{-1}-1)a_1 & -a_2^T \\ (2\gamma-1)a_2 & (2\omega^{-1}-1)A_3 \end{pmatrix} \end{aligned}$$

that

$$\frac{1}{2}\left\{(G-H)+(G-H)^T\right\} = \begin{pmatrix} (2\omega^{-1}-1)a_1 & (\gamma-1)a_2^T \\ (\gamma-1)a_2 & (2\omega^{-1}-1)A_3 \end{pmatrix}.$$

Therefore, by Lemma 3.1, it suffices to show that $(2\omega^{-1}-1)a_1 > 0$ and the Schur complement of $((G-H)+(G-H)^T)/2$ with respect to $(2\omega^{-1}-1)a_1$ is positive definite. In fact, we have $(2\omega^{-1}-1)a_1 > 0$ since $a_1 > 0$ from the positive definiteness of A and $0 < \omega < 2$ for all cases (a), (b) and (c). Moreover, the Schur complement of $((G-H)+(G-H)^T)/2$ with respect to $(2\omega^{-1}-1)a_1$ is written as

$$\begin{aligned} &(2\omega^{-1}-1)A_3 - \frac{(\gamma-1)^2a_2a_2^T}{(2\omega^{-1}-1)a_1} \\ &= \frac{1}{2\omega^{-1}-1}\left[\left\{(2\omega^{-1}-1)^2-(\gamma-1)^2\right\}A_3 + (\gamma-1)^2\left(A_3 - \frac{a_2a_2^T}{a_1}\right)\right], \end{aligned} \tag{3.6}$$

and it is positive definite by the following reason: Note that $2\omega^{-1}-1 > 0$ and the matrices A_3 and $A_3 - a_2a_2^T/a_1$ are positive definite. In case (b), we have $r - 1 = 0$ and $2\omega^{-1}-1 > 0$. In other cases, we have $(2\omega^{-1}-1)^2 - (\gamma-1)^2 \geq 0$ and $(\gamma-1)^2 > 0$ since $2\omega^{-1}-1 \geq \gamma-1 > 0$ in case (a) and $2\omega^{-1}-1 \geq 1-\gamma > 0$ in case (c). Hence, the matrix given by (3.6) is positive definite for all cases. ■

Proposition 3.1 *Suppose that $G \in \Re^{l\times l}$ is a positive definite matrix. Then, G is a $\mathcal{K}^l$-Q-matrix, and the solution of SOCCP$(p, G, \mathcal{K}^l)$ is unique for any $p \in \Re^l$.*

Proof. Let us define function F by $F(x) := Gx + p$. Then, F is strongly monotone since G is positive definite. Moreover, we note that SOCCP $(p, G, \mathcal{K}^l)$ is equivalent to the following variational inequality problem:

$$\begin{array}{rl} \text{Find} & x \in \mathcal{K}^l \\ \text{such that} & F(x)^T(y-x) \geq 0 \quad \forall y \in \mathcal{K}^l. \end{array}$$

Since any variational inequality problem with strongly monotone function has a unique solution [12, Corollary 3.2], we obtain the desired result. ■

Using the above lemmas and proposition, we give conditions for (3.1) with $B_{ii} = \Gamma(M_{ii}, \omega, \gamma)$ to be a regular $\mathcal{K}$-Q-splitting.

Theorem 3.1 *Let the splitting (B, C) of M be given by (3.1) with $B_{ii} = \Gamma(M_{ii}, \omega, \gamma)$, $i = 1, \dots, m$. Suppose that ω and γ satisfy either of the following conditions:*

$$\begin{array}{ll} \text{(a)} & \gamma > 1 \ \textit{and} \ 0 < \omega \leq 2/\gamma, \\ \text{(b)} & \gamma = 1 \ \textit{and} \ 0 < \omega < 2, \\ \text{(c)} & 0 \leq \gamma < 1 \ \textit{and} \ 0 < \omega \leq 2/(2-\gamma). \end{array}$$

Then (B, C) is a regular $\mathcal{K}$-Q-splitting.

Proof. Let (B_{ii}, C_{ii}) be a splitting of M_{ii} where $B_{ii} = \Gamma(M_{ii}, \omega, \gamma)$ and $C_{ii} = M_{ii} - B_{ii}$. First we show that, for each $i = 1, \dots, m$, B_{ii} is positive definite and the splitting (B_{ii}, C_{ii}) is regular. Note that either condition (i) or (ii) in Lemma 3.2 holds since $2/(2-\gamma) < 2 < 2/\gamma$ when $0 < \gamma < 1$ and $2/(2-\gamma) = 2 = 2/\gamma$ when $\gamma = 1$. Moreover, since M_{ii} is symmetric and positive definite, we obtain the positive definiteness of B_{ii} by setting $A := M_{ii}$ and $G := B_{ii} = \Gamma(M_{ii}, \omega, \gamma)$ in Lemma 3.2. Similarly, we obtain the regularity of the splitting (B_{ii}, C_{ii}), by setting $A := M_{ii}$, $G := B_{ii} = \Gamma(M_{ii}, \omega, \gamma)$ and $H := C_{ii} = M_{ii} - \Gamma(M_{ii}, \omega, \gamma)$ in Lemma 3.3.

Using these results, we show that (B, C) is a regular $\mathcal{K}$-Q-splitting. We first show that B is a $\mathcal{K}$-Q-matrix. Since B_{ii} is positive definite, we have from Proposition 3.1 that the matrix B_{ii} is also a $\mathcal{K}^{n_i}$-Q-matrix. Note, moreover, that the solution of SOCCP (2.2) is obtained by solving SOCCP (3.2) recursively from $i = 1$ to $i = m$, and that each SOCCP is solvable since B_{ii} is a $\mathcal{K}^{n_i}$-Q-matrix. Hence, the whole SOCCP $(B, q, \mathcal{K})$ is solvable for any q, that is, B is a $\mathcal{K}$-Q-matrix. We next show the regularity of (B, C). Since M is symmetric and B is block lower triangular, $B - C$ can be written as

$$B - C = \operatorname{diag}\{B_{ii} - C_{ii}\}_{i=1}^m + L - L^T,$$

where $\operatorname{diag}\{B_{ii} - C_{ii}\}_{i=1}^m$ denotes a block diagonal matrix whose diagonal elements are $B_{ii} - C_{ii}$, and $L \in \Re^{n\times n}$ is the strictly block lower triangular part of M. Then, we have for any $z \in \Re^n \backslash \{0\}$

$$\begin{aligned} z^T(B-C)z &= z^T(\operatorname{diag}\{B_{ii} - C_{ii}\}_{i=1}^m)z + z^T L z - z^T L^T z \\ &= \sum_{i=1}^m {z_i}^T (B_{ii} - C_{ii}) z_i \\ &> 0, \end{aligned}$$

where the inequality follows from the regularity of (B_{ii}, C_{ii}). Hence, the splitting (B, C) is regular. ■

4 Solving subproblems

In the previous section, we have shown that SOCCP (2.2) can be decomposed into m subproblems (3.2) by choosing the splitting (B, C) as in (3.1). We have also derived conditions for Algorithm 2.1 with the splitting (3.1) to be convergent. In this section, we propose a method for solving these m subproblems (3.2) efficiently. In order to simplify the notation, we consider the following SOCCP in which superscripts and subscripts are omitted:

$$\begin{array}{rl} \text{Find} & z \in \Re^l \\ \text{such that} & z \in \mathcal{K}^l, \;\; Bz + r \in \mathcal{K}^l, \;\; z^T(Bz + r) = 0, \end{array} \tag{4.1}$$

where B is a positive definite matrix of the form

$$B = \begin{pmatrix} b_1 & 0^T \\ b_2 & B_3 \end{pmatrix} \tag{4.2}$$

with $b_1 \in \Re$, $b_2 \in \Re^{l-1}$ and a symmetric matrix $B_3 \in \Re^{(l-1)\times(l-1)}$. Notice that each subproblem (3.2) reduces to SOCCP (4.1) when the parameters ω and γ used in the splitting (3.1) with $B_{ii} = \Gamma(M_{ii}, \omega, \gamma)$ satisfy the assumptions of Theorem 3.1. In addition, SOCCP (4.1) has a unique solution from the positive definiteness of B and Proposition 3.1. When $l = 1$, we can easily obtain the solution of (4.1) as $z = \max(0, -r/B)$. So we will consider the case $l \geq 2$.

The following three cases are possible for a solution z^* of SOCCP (4.1):

(i) $z^* = 0$,

(ii) $z^* \in \operatorname{int} \mathcal{K}^l$,

(iii) $z^* \in \operatorname{bd} \mathcal{K}^l \setminus \{0\}$,

where $\operatorname{int} \mathcal{K}^l$ and $\operatorname{bd} \mathcal{K}^l$ denote the interior and the boundary of $\mathcal{K}^l$, respectively. Since $z^* \in \mathcal{K}^l$, it is clear that no other cases are possible for a solution of (4.1). To solve SOCCP (4.1) efficiently, it will be helpful to detect which case applies to the solution z^*. To this end, we provide the following proposition.

Proposition 4.1 *Let z^* be the unique solution of SOCCP (4.1). Then,*

(a) *case* (i) *holds if and only if $r \in \mathcal{K}^l$;*

(b) *case* (ii) *holds if and only if $-B^{-1}r \in \operatorname{int} \mathcal{K}^l$. Moreover, we have $z^* = -B^{-1}r$;*

(c) *case* (iii) *holds if and only if $r \notin \mathcal{K}^l$ and $-B^{-1}r \notin \operatorname{int} \mathcal{K}^l$.*

Proof. We first show (a). If $z^* = 0$ solves SOCCP (4.1), then we have $Bz^* + r = r \in \mathcal{K}^l$. Conversely, if $r \in \mathcal{K}^l$, it is easily seen that $z^* = 0$ solves SOCCP (4.1). We next show (b). If case (ii) holds, then we have $Bz^* + r = 0$, which implies $z^* = B^{-1}r$. Conversely, if $-B^{-1}r \in \operatorname{int} \mathcal{K}^l$, it is easily seen that $z^* = -B^{-1}r$ solves SOCCP (4.1). We must have (c) from the existence and the uniqueness of the solution. This completes the proof. ∎

This proposition indicates that, if $r \in \mathcal{K}^l$ or $-B^{-1}r \in \operatorname{int} \mathcal{K}^l$, then we can readily calculate the solution of SOCCP (4.1).

Now, we describe a method of finding the solution of SOCCP (4.1) when case (iii) holds. Note that we have $Bz^* + r \in \operatorname{bd} \mathcal{K}^l$ since the solution z^* belongs to $\operatorname{bd} \mathcal{K}^l \setminus \{0\}$ and the inner product of z^* and $Bz^* + r$ is equal to 0. Thus, we can write z^* and $Bz^* + r$ as

$$z^* = \lambda \begin{pmatrix} 1 \\ \tilde{w} \end{pmatrix}, \tag{4.3}$$

$$Bz^* + r = \mu \begin{pmatrix} 1 \\ -\tilde{w} \end{pmatrix}, \tag{4.4}$$

where $\lambda > 0$, $\mu \geq 0$ and $\tilde{w}$ is an $(l-1)$-dimensional vector such that $\|\tilde{w}\| = 1$. Generally, it is not easy to find z^* satisfying (4.3) and (4.4) simultaneously. Nevertheless, when B is given by (4.2), we can exploit the special structure of B to compute the solution z^* as follows: Substituting (4.2) and (4.3) into (4.4), we have

$$\begin{pmatrix} \lambda b_1 + r_1 \\ \lambda b_2 + \lambda B_3 \tilde{w} + r_2 \end{pmatrix} = \begin{pmatrix} \mu \\ -\mu \tilde{w} \end{pmatrix}, \tag{4.5}$$

where $r = (r_1, r_2) \in \Re \times \Re^{l-1}$. Eliminating μ in (4.5), we have

$$-\{\lambda(b_1 I + B_3) + r_1 I\}\tilde{w} = \lambda b_2 + r_2. \tag{4.6}$$

Moreover, $\lambda b_1 + r_1 = \mu$ together with $\lambda > 0$, $\mu \geq 0$ and $b_1 > 0$ from $B \succ O$ yields

$$\lambda \geq \lambda_L := \max\{0, -r_1/b_1\}.$$

Let us define function $H : \Re \to \Re^{(l-1)\times(l-1)}$ by

$$H(\lambda) := \lambda(b_1 I + B_3) + r_1 I. \tag{4.7}$$

If $H(\lambda)$ is nonsingular, then (4.6) can be rewritten as

$$\tilde{w} = -H(\lambda)^{-1}(\lambda b_2 + r_2). \tag{4.8}$$

Thus, if we find a λ^* such that $\|H(\lambda^*)^{-1}(\lambda^* b_2 + r_2)\| = 1$ and $\lambda^* \geq \lambda_L$, then we obtain z^* by (4.3) and (4.8).

If $r_1 = 0$, then the equation $\|H(\lambda)^{-1}(\lambda b_2 + r_2)\| = 1$, together with $H(\lambda) = \lambda(b_1 I + B_3)$, yields the single variable quadratic equation $\lambda^2(1 - \|g\|^2) - 2\lambda g^T h - \|h\|^2 = 0$, where $g := (b_1 I + B_3)^{-1} b_2$ and $h := (b_1 I + B_3)^{-1} r_2$. Since $\lambda^* \geq \lambda_L = 0$ and $\|g\| < 1$ from the following lemma, λ^* is given by

$$\lambda^* = \frac{-g^T h + \sqrt{(g^T h)^2 + \|h\|^2(1 - \|g\|^2)}}{1 - \|g\|^2}. \tag{4.9}$$

Lemma 4.1 *Let B be an arbitrary positive definite matrix given by* (4.2)*, and* $g := (b_1 I + B_3)^{-1} b_2$*. Then, we have* $\|g\| < 1$*.*

Proof. Let $v := (1, -g^T)^T$. Then we have

$$\begin{aligned} v^T B v &= b_1 - g^T b_2 + g^T B_3 g \\ &= b_1 - g^T(b_1 I + B_3)g + g^T B_3 g \\ &= b_1(1 - \|g\|^2). \end{aligned}$$

Since $v \neq 0$ and B is positive definite, we have $b_1 > 0$ and $b_1(1 - \|g\|^2) > 0$, which imply $\|g\| < 1$. ■

Since z^* can be easily obtained by (4.9), (4.8) and (4.3) when $r_1 = 0$, we suppose $r_1 \neq 0$ in the subsequent discussions. The following proposition gives a sufficient condition for $H(\lambda)$ to be nonsingular.

Proposition 4.2 *If $r_1 \neq 0$, then the matrix $H(\lambda)$ defined by* (4.7) *is positive definite for any $\lambda \geq \lambda_L$.*

Proof. Note that $b_1 > 0$ and B_3 is positive definite since B is positive definite. If $r_1 > 0$, then $\lambda_L = 0$ and hence $H(\lambda) = \lambda(b_1 I + B_3) + r_1 I$ is positive definite for any $\lambda \geq \lambda_L$. If $r_1 < 0$, then $\lambda_L = -r_1/b_1 > 0$ and hence $H(\lambda) = \lambda B_3 + b_1(\lambda + r_1/b_1)I$ is positive definite for any $\lambda \geq \lambda_L$. ■

Now, let us define functions $w : [\lambda_L, +\infty) \to \Re^{l-1}$ and $\psi : [\lambda_L, +\infty) \to \Re$ by

$$w(\lambda) := -H(\lambda)^{-1}(\lambda b_2 + r_2), \tag{4.10}$$
$$\psi(\lambda) := \|w(\lambda)\|. \tag{4.11}$$

Then, our purpose is to find a solution $\lambda^* \geq \lambda_L$ of the following single variable equation:

$$\psi(\lambda) = 1. \tag{4.12}$$

First we confirm that such a solution always exists.

Proposition 4.3 *Suppose that case* (iii) *holds and* $r_1 \neq 0$. *Then we have* (a) $\psi(\lambda_L) \geq 1$ *and* (b) $\lim_{\lambda \to +\infty} \psi(\lambda) = \|(b_1 I + B_3)^{-1} b_2\| < 1$.

Proof. Since (b) can be easily obtained from (4.10) and Lemma 4.1, we only show (a). Assume to contrary that $\psi(\lambda_L) < 1$. Then, it suffices to show that either case (i) or (ii) holds, since the solution exists uniquely. When $r_1 > 0$, we have from $w(\lambda_L) = w(0) = -r_1^{-1} r_2$ that $1 > \psi(\lambda_L) = \|w(0)\| = \|r_2\|/r_1$. This implies $r \in \operatorname{int} \mathcal{K}^l \subset \mathcal{K}^l$, that is, case (i) holds. When $r_1 < 0$, (4.10) and (4.11), together with $b_1 > 0$, $r_1 < 0$ and $\lambda_L = -r_1 b_1^{-1}$, yield

$$\psi(\lambda_L) = \frac{\| - B_3^{-1}(r_2 - r_1 b_1^{-1} b_2)\|}{-r_1 b_1^{-1}} < 1,$$

which implies

$$\begin{pmatrix} -r_1 b_1^{-1} \\ -B_3^{-1}(r_2 - r_1 b_1^{-1} b_2) \end{pmatrix} \in \operatorname{int} \mathcal{K}^l. \tag{4.13}$$

The vector in (4.13) equals $-B^{-1} r$ since the following equality holds identically:

$$\begin{pmatrix} b_1 & 0^T \\ b_2 & B_3 \end{pmatrix} \begin{pmatrix} -r_1 b_1^{-1} \\ -B_3^{-1}(r_2 - r_1 b_1^{-1} b_2) \end{pmatrix} = -\begin{pmatrix} r_1 \\ r_2 \end{pmatrix}.$$

Hence, we have $-B^{-1} r \in \operatorname{int} \mathcal{K}^l$, that is, case (ii) holds. ∎

Since function ψ is continuous for $\lambda \geq \lambda_L$, Proposition 4.3 guarantees the existence of $\lambda^* \geq \lambda_L$ satisfying (4.12). Actually, such a λ^* must exist uniquely since λ^* gives a solution of SOCCP (4.1), which exists uniquely from the positive definiteness of B and Proposition 3.1.

Now, we present a procedure for finding a solution $\lambda^* \geq \lambda_L$ of equation (4.12), which is a Newton type method incorporating the bisection method as a safeguard strategy. In particular, instead of applying Newton's method to (4.12) directly, we will adopt a more efficient method that is reminiscent of an approach well-known in the trust region literature [6, Chapter 7].

Define function $\phi : [\lambda_L, +\infty) \to (-1, +\infty]$ by

$$\phi(\lambda) = \begin{cases} \psi(\lambda)^{-1} - 1 & \psi(\lambda) > 0 \\ +\infty & \text{if } \psi(\lambda) = 0. \end{cases}$$

Then, the nonlinear equation

$$\phi(\lambda) = 0 \tag{4.14}$$

is equivalent to (4.12). We aim to solve (4.14) instead of (4.12) since function ϕ is expected to behave better than ψ [6] .

The iterative formula of Newton's method for the nonlinear equation (4.14) is given by

$$\lambda_{j+1} = \lambda_j - \frac{\phi(\lambda_j)}{\phi'(\lambda_j)}, \tag{4.15}$$

where ϕ' denotes the derivative of ϕ. The derivative ϕ' is obtained by

$$\begin{aligned} \phi'(\lambda) &= \left\{ \frac{1}{\|w(\lambda)\|} - 1 \right\}' \\ &= -\frac{w(\lambda)^T w'(\lambda)}{\|w(\lambda)\|^3} \\ &= \frac{w(\lambda)^T H(\lambda)^{-1}\{(b_1 I + B_3)w(\lambda) + b_2\}}{\|w(\lambda)\|^3}, \end{aligned} \tag{4.16}$$

where the last equality follows from

$$(b_1 I + B_3)w(\lambda) + H(\lambda)w'(\lambda) \;=\; -b_2,$$

which is obtained by differentiating both sides of $H(\lambda)w(\lambda) = -(\lambda b_2 + r_1)$. Since $H(\lambda)$ is positive definite for any $\lambda \geq \lambda_L$, by using the Cholesky factorization $H(\lambda) = R(\lambda)R(\lambda)^T$ with $R(\lambda)$ being upper triangular [10], we may rewrite the formula (4.15) as follows:

$$\begin{aligned} \lambda_{j+1} &= \lambda_j - \left(\frac{1}{\|w(\lambda_j)\|} - 1 \right) \frac{\|w(\lambda_j)\|^3}{w(\lambda_j)^T (R(\lambda_j)^T)^{-1} R(\lambda_j)^{-1} \{(b_1 I + B_3) w(\lambda_j) + b_2\}} \\ &= \lambda_j + (\|w(\lambda_j)\| - 1) \frac{\|w(\lambda_j)\|^2}{u(\lambda_j)^T v(\lambda_j)}, \end{aligned}$$

where $u(\lambda) = R(\lambda)^{-1} w(\lambda)$ and $v(\lambda) = R(\lambda)^{-1}\{(b_1 I + B_3)w(\lambda) + b_2\}$. Summarizing the above arguments, we have the following procedure for solving SOCCP (4.1).

Procedure 4.1

Step 1. *If* $r \in \mathcal{K}^l$ (case (i)), *set* $z^* := 0$ *and terminate.*

Step 2. *If* $-B^{-1}r \in \operatorname{int} \mathcal{K}^l$ (case (ii)), *set* $z^* := -B^{-1}r$ *and terminate.*

Step 3. *Otherwise* (case (iii)), *calculate* z^* *as follows:*

- **Step 3-0.** *If* $r_1 \neq 0$, *then go to Step* 3-1. *Otherwise, calculate vectors* g *and* h *such that* $(b_1 I + B_3)g = b_2$ *and* $(b_1 I + B_3)h = r_2$. *Set* $\lambda^* := (-g^T h + ((g^T h)^2 + \|h\|^2(1 - \|g\|^2))^{1/2})/(1 - \|g\|^2)$ *and* $z^* := \lambda^*(1, w(\lambda^*)^T)^T$. *Terminate.*
- **Step 3-1.** *Let* $\lambda_0 := \lambda_L$, $\alpha_0 := \lambda_L$ *and* β_0 *be a scalar such that* $\beta_0 > \lambda_L$ *and* $\|w(\beta_0)\| < 1$. *Set* $j := 0$.
- **Step 3-2.** *Factorize* $H(\lambda_j) = RR^T$. *Let* $R^j := R$.
- **Step 3-3.** *Calculate* w^j *such that* $R^j (R^j)^T w^j = -(\lambda_j b_2 + r_2)$.
- **Step 3-4.** *If* $\|w^j\| = 1$, *then set* $z^* := \lambda_j (1, (w^j)^T)^T$ *and terminate. If* $\|w^j\| > 1$, *then let* $\alpha_{j+1} := \lambda_j$ *and* $\beta_{j+1} := \beta_j$. *If* $\|w^j\| < 1$, *then let* $\alpha_{j+1} := \alpha_j$ *and* $\beta_{j+1} := \lambda_j$.
- **Step 3-5.** *Calculate* u^j *such that* $R^j u^j = w^j$.
- **Step 3-6.** *Calculate* v^j *such that* $R^j v^j = (b_1 I + B_3) w^j + b_2$.
- **Step 3-7.** *Let* $\tilde{\lambda}_{j+1} := \lambda_j + (\|w^j\| - 1)\|w^j\|^2/((u^j)^T v^j)$. *If* $\tilde{\lambda}_{j+1} \in (\alpha_{j+1}, \beta_{j+1})$, *then let* $\lambda_{j+1} := \tilde{\lambda}_{j+1}$. *Otherwise, let* $\lambda_{j+1} := (\alpha_{j+1} + \beta_{j+1})/2$. *Set* $j := j + 1$ *and go back to* Step 3-2.

This procedure is used in Step 2 of Algorithm 2.1 to solve subproblem (3.2). In practice, we may set $\beta_0 := +\infty$, and then compute $\beta_j < +\infty$ such that $\|w(\beta_j)\| < 1$ only when it becomes necessary for the first time.

Since a bisection-type safeguard strategy is combined with Newton's method, Procedure 4.1 is guaranteed to be globally convergent. Note that, when the parameter γ is chosen to be 0 in the splitting (3.1) with $B_{ii} = \Gamma(M_{ii}, \omega, \gamma)$, SOCCP (3.2) reduces to SOCCP (4.1) with $b_2 = 0$, and the convergence is ensured even without using the safeguard strategy. In fact, since $H(\lambda)^{-1}(b_1 I + B_3) = \lambda I + r_1(b_1 I + B_3)^{-1}$ is positive definite for any $\lambda \geq \lambda_L$, it follows from (4.16) with $b_2 = 0$ that $\phi'(\lambda) > 0$, and hence, function ϕ is monotonically increasing on the interval $[\lambda_L, +\infty)$. We can further show that ϕ is concave on the interval $[\lambda_L, +\infty)$ by calculating the second derivative ϕ'' of ϕ. Thus, starting from $\lambda_0 := \lambda_L$, Newton's method generates a monotonically increasing sequence $\{\lambda_j\}$ converging to a solution λ^*. Furthermore, since ϕ is almost linear for $\lambda \geq \lambda_L$, we may expect that Newton's method converges very rapidly [6].

On the other hand, when $\gamma \neq 0$, convergence is not guaranteed unless a safeguard strategy is employed, since $b_2 \neq 0$ in (4.16) and hence ϕ is generally neither monotonically increasing nor concave on the interval $[\lambda_L, +\infty)$. Nevertheless, it is worth mentioning that, as is shown in Theorem 3.1, it is allowed to choose an ω such that $\omega > 1$ when $0 < \gamma 2$, whereas ω must be in $(0, 1]$ when $\gamma = 0$. In view of the fact that convergence of SOR method for linear complementarity problems is accelerated by letting $\omega > 1$ [8], we may expect that Algorithm 2.1 performs better when ω is chosen greater than 1.

5 Numerical results

In this section, we present some numerical results with the proposed algorithm. The program was coded in MATLAB 6.5.0 and run on a computer with 3.04GHz CPU and 2GB memory. We have conducted the following experiments:

(A) Testing Procedure 4.1 on SOCCP (4.1) of various problem sizes.

(B) Testing Procedure 4.1 on SOCCP (4.1) with various degrees of sparsity.

(C) Testing Algorithm 2.1 on SOCCP (1.1) with various Cartesian structures of $\mathcal{K}$.

(D) Testing Algorithm 2.1 on SOCCP (1.1) with various values of γ and ω.

In experiments (A)−(C), we used not only complete Cholesky factorization but also incomplete Cholesky factorization in Step 3-2 of Procedure 4.1. The complete Cholesky factorization provides upper triangular R satisfying $H(\lambda) = RR^T$ exactly. But, even if matrix $H(\lambda)$ is sparse, R is not sparse in general. On the other hand, the incomplete Cholesky factorization provides upper triangular R such that $H(\lambda) \approx RR^T$. Therefore, $\|w^j\| = \| -(R^j(R^j)^T)^{-1}(\lambda_j b_2 + r_2)\| = 1$ does not necessarily imply $\|w(\lambda_j)\| = \| -H(\lambda_j)^{-1}(\lambda_j b_2 + r_2)\| = 1$, and the point z^* obtained by Procedure 4.1 may only be an approximate solution of SOCCP (4.1). However, since the incomplete Cholesky factorization exploits the sparsity of a matrix, computational cost can be reduced drastically when $H(\lambda)$ is sparse. In addition, our computational experiments has revealed that, even though subproblems (2.2) are solved only approximately, Algorithm 2.1 is able to find a solution of SOCCP (1.1) in most cases.

In experiment (A), we generated 100 test problems for each $l = 100, 200, \dots, 1000$ and solved each problem by Procedure 4.1 with complete Cholesky factorization and incomplete Cholesky factorization. The termination criterion was $|\|w^j\| - 1| < 10^{-4}$. In generating a test problem, elements of vector r were chosen randomly from the interval $[-1, 1]$, and a positive definite matrix B of the form (4.2) was obtained by the following procedure: First, set $A = NN^T + D$, where N is a square matrix whose nonzero elements are chosen randomly from the interval $[-1, 1]$, and D is a diagonal matrix whose diagonal elements are chosen randomly from $[0.01, 1]$. Then, let $B := \Gamma(A, 1, 2)$, where Γ is defined by (3.3). In this procedure, the number of nonzero elements of N is determined so that the nonzero density of matrix B becomes approximately 5%. We show the results in Table 1, where l denotes the number of variables, $\sharp$iter denotes the number of iterations, and cpu(s) denotes the CPU time in second. The left column labeled comChol corresponds to the case where the complete Cholesky factorization was employed, while the right column labeled incChol corresponds to the case where the incomplete Cholesky factorization was employed. In particular, the number of iterations and the CPU time are the averages of 100 trials for each l. We may observe that the number of iterations stays almost constant regardless of the problem size, although the CPU time grows with the problem size.

Table 1: Results for Procedure 4.1 applied to SOCCPs (4.1) of various problem sizes

l	comChol		incChol	
	$\sharp$iter	cpu(s)	$\sharp$iter	cpu(s)
100	4.40	0.008	4.53	0.006
200	4.70	0.048	4.84	0.022
300	4.96	0.179	5.12	0.057
400	4.99	0.436	5.33	0.125
500	5.09	0.872	5.28	0.232
600	5.25	1.707	5.51	0.407
700	5.18	2.555	5.36	0.619
800	5.29	3.809	5.36	0.932
900	5.26	5.362	5.48	1.343
1000	5.31	7.264	5.65	1.875

In experiment (B), we generated 100 test problems for each nonzero density 0.2%, 0.5%, 1%, 2%, 5% and 10%, where the problem size l was fixed to 1000. The termination criterion and the procedure for generating B and r were similar to experiment (A). We show the results in Table 2, where dens denotes the (approximate) nonzero density of matrix B, $\sharp$iter denotes the number of iterations, and cpu(s) denotes the CPU time in second. Each number is the average of 100 trials for each nonzero density. As shown in Table 2, the sparsity of matrix B affects the number of iterations only slightly, but reduces the computational cost drastically.

Table 2: Results for Procedure 4.1 applied to SOCCPs (4.1) with various degrees of sparsity

dens	comChol		incChol	
	$\sharp$iter	cpu(s)	$\sharp$iter	cpu(s)
0.2%	3.89	0.024	4.01	0.055
0.5%	4.54	0.561	4.47	0.118
1%	4.83	2.617	4.85	0.277
2%	5.12	5.085	5.12	0.682
5%	5.32	7.531	5.52	1.858
10%	5.43	8.952	5.80	3.533

In experiment (C), we solved SOCCP (1.1) with various Cartesian structures of $\mathcal{K}$, by incorporating a block SOR method and Procedure 4.1 into Algorithm 2.1. To construct SOCs of various types, we chose n_i and m such that $n_1 + \cdots + n_m = 1600$ and $n_1 = \cdots = n_m$, where m is the number of

SOCs comprising $\mathcal{K}$ and n_i is the dimension of each SOC. For each type of $\mathcal{K}$, we solved 100 test problems with data (M, q) randomly generated as follows: Elements of vector q were chosen from the interval $[-1, 1]$, and a symmetric positive definite matrix M, whose nonzero density was controlled to be approximately 1%, was given by $M = NN^T + D$, where N and D are matrices obtained in a way similar to experiment (A). We set $\gamma = 1$ and $\omega = 1.1$ and let the initial point be $z^0 = 0$. The termination criterion was $\|z^{k+1} - z^k\| < 10^{-8}$. We show the results in Table 3, where the number of iterations and the CPU time are the averages of 100 trials for each type of $\mathcal{K}$. We find that the number of iterations gradually decreases as the number m of SOCs in $\mathcal{K}$ decreases. However, the whole CPU time does not decrease monotonically as m decreases, since subproblems to be solved at each iteration become more expensive as n_i becomes larger.

Table 3: Results for Algorithm 2.1 applied to SOCCPs (1.1) with various Cartesian structures of $\mathcal{K}$

n_i	m	comChol		incChol	
		♯iter	cpu(s)	♯iter	cpu(s)
1	1600	36.00	7.814	35.02	7.528
2	800	28.25	10.244	27.40	9.662
5	320	23.05	6.562	23.85	6.609
10	160	21.85	4.355	22.00	4.305
20	80	21.10	2.892	21.28	2.890
40	40	21.32	2.144	20.70	2.094
80	20	19.77	1.481	19.87	1.532
160	10	19.34	1.290	19.16	1.355
320	5	17.27	2.988	17.67	1.658
800	2	14.95	36.988	14.63	4.594
1600	1	11.99	175.595	13.13	19.179

In experiment (D), we solved SOCCP (1.1) with various values of parameters γ and ω used to determine the splitting (B, C). Specifically, we tried all possible combinations of $(\omega, \gamma) \in \{0.1, 0.2, \ldots, 1.9\} \times \{0, 0.5, 1.0, 1.5, 2.0\}$. The underlying SOC $\mathcal{K}$ was fixed to be $\mathcal{K}^{20} \times \cdots \times \mathcal{K}^{20} \subset \Re^{400}$. We solved different 100 test problems for each pair (ω, γ), where M and q were generated in a way similar to experiment (C). The termination criterion was $\|z^{k+1} - z^k\| < 10^{-8}$, and the complete Cholesky factorization was used in Step 3-2 of Procedure 4.1. We show the results in Table 4. The numbers in the table show the average numbers of iterations taken over the successful trials for each pair (ω, γ). We see that the algorithm converges most rapidly when $\omega = 1.1$ for every γ. Note that the combinations of (ω, γ) marked with parentheses in the table do not satisfy the assumptions of Theorem 3.1, and hence, the convergence is not guaranteed for such (ω, γ). In fact, the numbers in the parentheses show how often the algorithm with such (ω, γ) failed to converge. Nevertheless, we were able to obtain the solution in most instances even if the convergence of the algorithm was not assured theoretically.

6 Concluding Remarks

In this paper, we have extended the matrix splitting method for LCPs to SOCCPs and showed that the algorithm converges under the assumption that the matrix M is strictly $\mathcal{K}$-copositive. Furthermore, we have proposed a block SOR method for solving SOCCPs with M being positive definite, and demonstrated its effectiveness through numerical experiments. From a practical viewpoint, there is room for improvement in speeding up the algorithm. In particular, it would be highly effective if

Table 4: Results for Algorithm 2.1 applied to SOCCPs (1.1) with various choices of (ω, γ)

$\omega \backslash \gamma$	0	0.5	1.0	1.5	2.0
0.1	321.61	323.26	335.16	323.47	336.67
0.2	164.24	169.03	160.09	161.46	164.66
0.3	104.50	104.59	106.34	104.08	104.13
0.4	76.63	77.89	76.73	78.18	78.40
0.5	59.71	57.71	59.35	59.51	59.88
0.6	47.22	45.77	46.62	45.41	45.34
0.7	38.54	37.13	38.10	37.40	35.84
0.8	30.02	30.17	29.02	28.35	29.19
0.9	25.95	24.30	23.68	23.00	22.98
1.0	20.32	19.54	18.30	18.73	19.72
1.1	17.88(0)	17.65	16.97	17.25	18.33(0)
1.2	18.75(0)	18.45	18.41	19.07	20.31(0)
1.3	21.30(0)	20.92	21.21	21.47	22.62(0)
1.4	25.38(0)	24.38(0)	24.43	25.18(0)	26.34(2)
1.5	31.39(0)	30.34(0)	29.53	31.70(0)	31.19(2)
1.6	41.78(2)	37.93(0)	37.81	39.51(0)	39.82(0)
1.7	53.98(0)	51.93(0)	51.99	54.82(1)	55.77(6)
1.8	81.59(2)	77.21(0)	73.64	75.50(1)	79.71(6)
1.9	135.04(2)	147.93(1)	128.84	128.33(2)	152.04(14)

subproblems are solved in parallel. Moreover, it is desirable to develop a matrix splitting method for SOCCPs that is convergent under weaker conditions.

References

[1] F. Alizadeh, D. Goldfarb, Second-order cone programming, Mathematical Programming, 95 (2003), 3–51.

[2] C. Chen, O.L. Mangasarian, Smoothing methods for convex inequalities and linear complementarity problems, Mathematical Programming, 71 (1995), 51–69.

[3] J.-S. Chen, X. Chen, P. Tseng, Analysis of nonsmooth vector-valued functions associated with second-order cones, manuscript, Department of Mathematics, University of Washington, December 2002.

[4] X. Chen, Smoothing methods for complementarity problems and their applications: A survey, Journal of the Operations Research of Japan, 43 (2000), 32–47.

[5] X.D. Chen, D. Sun, J. Sun, Complementarity functions and numerical experiments for second-order-cone complementarity problems, Computational Optimization and Applications, (to appear).

[6] A.R. Conn, N.I.M. Gould, P.L. Toint, Trust-Region Methods, Society for Industrial and Applied Mathematics, Philadelphia, 2000.

[7] R.W. Cottle, J.-S. Pang, R.E. Stone, The Linear Complementarity Problem, Academic Press, San Diego, 1992.

[8] R. De Leone, O.L. Mangasarian, Asynchronous parallel successive overrelaxation for the symmetric linear complementarity problem, Mathematical Programming, 42 (1988), 347–361.

[9] M. Fukushima, Z.-Q. Luo, P. Tseng, Smoothing functions for second-order cone complementarity problems, SIAM Journal on Optimization, 12 (2001), 436–460.

[10] G.H. Golub, C.F.V. Loan, Matrix Computations, The Johns Hopkins University Press, Baltimore, 1996.

[11] G.H. Golub, J.M. Ortega, Scientific Computing: An Introduction with Parallel Computing, Academic Press, San Diego, 1993.

[12] P.T. Harker, J.-S. Pang, Finite-dimensional variational inequality and nonlinear complementarity problems : A survey of theory, algorithms and applications, Mathematical Programming, 48 (1990), 161–220.

[13] S. Hayashi, N. Yamashita, M. Fukushima, A combined smoothing and regularization method for monotone second-order cone complementarity problems, Tech. Rep. 2003-002, Department of Applied Mathematics and Physics, Kyoto University, January 2003.

[14] R.A. Horn, C.R. Johnson, Matrix Analysis, Cambridge University Press, Cambridge, 1985.

[15] A.N. Iusem, On the convergence of iterative methods for symmetric linear complementarity problems, Mathematical Programming, 59 (1993), 33–48.

[16] C. Kanzow, H. Pieper, Jacobian smoothing methods for general complementarity problems, SIAM Journal on Optimization, 9 (1999), 342–373.

[17] M.S. Lobo, L. Vandenberghe, S. Boyd, H. Lebret, Applications of second-order cone programming, Linear Algebra and Its Applications, 284 (1998), 193–228.

[18] Z.-Q. Luo, P. Tseng, Error bound and convergence analysis of matrix splitting algorithms for the affine variational inequality problem, SIAM Journal on Optimization, 2 (1992), 43–54.

[19] N. Machida, M. Fukushima, T. Ibaraki, A multisplitting method for symmetric linear complementarity problems, Journal of Computational and Applied Mathematics, 62 (1995), 217–227.

[20] O.L. Mangasarian, R. De Leone, Parallel successive overrelaxation methods for symmetric linear complementarity problems, Journal of Optimization Theory and Applications, 54 (1987), 437–446.

[21] L. Qi, D. Sun, Smoothing functions and a smoothing Newton method for complementarity and variational inequality problems, Journal of Optimization Theory and Applications, 113 (2002), 121–147.

[22] J.P. Zeng, D.H. Li, M. Fukushima, Weighted max-norm estimate of additive Schwarz iteration scheme for solving linear complementarity problems, Journal of Computational and Applied Mathematics, 131 (2001), 1–14.