

Computation Algorithm for Convex Semi-infinite Program with Second-Order Cones: Special Analyses for Affine and Quadratic Case

Shunsuke Hayashi¹ · Soon-Yi Wu^{2,3} · Liping Zhang⁴

Received: 9 September 2014 / Revised: 1 July 2015 / Accepted: 30 November 2015 /
Published online: 15 December 2015
© Springer Science+Business Media New York 2015

Abstract We focus on the convex semi-infinite program with second-order cone constraints (for short, SOCCSIP), which has wide applications such as filter design, robust optimization, and so on. For solving the SOCCSIP, we propose an explicit exchange method, and prove that the algorithm terminates in a finite number of iterations. In the convergence analysis, we do not need to use the special structure of second-order cone (SOC) when the objective or constraint function is strictly convex. However, if both of them are non-strictly convex and constraint function is affine or quadratic, then we have to utilize the SOC complementarity conditions and the spectral factorization techniques associated with Euclidean Jordan algebra. We also show that the obtained output is an approximate optimum of SOCCSIP. We report some numerical results involving the application to the robust optimization in the classical convex semi-infinite program.

Keywords Continuous optimization · Semi-infinite program · Exchange method · Second-order cone

Mathematics Subject Classification 90C30 · 90C33

This research was supported in part by JSPS KAKENHI Grant Number 26330022, and by National Center for Theoretical Sciences, Taiwan.

✉ Soon-Yi Wu
soonyi@mail.ncku.edu.tw

Shunsuke Hayashi
s_hayashi@plan.civil.tohoku.ac.jp

Liping Zhang
lzhang@math.tsinghua.edu.cn

¹ Graduate School of Information Sciences, Tohoku University, Sendai 980-8579, Japan

² Department of Mathematics, National Cheng Kung University, Tainan 70101, Taiwan

³ National Center for Theoretical Sciences, Tainan, Taiwan

⁴ Department of Mathematical Sciences, Tsinghua University, Beijing 100084, China

1 Introduction

The semi-infinite program (SIP) [11, 16, 29] is a constrained optimization with the feasible region defined by means of infinitely many inequality constraints. This problem has strong practical backgrounds in engineering problems such as optimal control, optimal filter design in signal processing, resource allocation in decentralized systems, decision making under competition, least square problems, equilibrium problems, and so on [12, 23, 35, 36]. However, the SIP is much more difficult than the standard nonlinear program (NLP) since the infinitely many inequality constraints have to be dealt with.

So far, many numerical algorithms have been proposed for solving SIPs (see [18, 21, 28, 30] and references therein). A typical algorithm for solving an SIP generates a sequence of finitely constrained subproblems that can be solved by some standard algorithms for NLPs. Existing methods for SIPs can be roughly divided into three types: exchange methods [19, 28], discretization methods, and reduction-based methods (see, e.g. [28, 31] and references therein). The exchange and the discretization methods are numerically expensive in general since the cost per iteration increases dramatically as the cardinality of the auxiliary problem grows. The reduction-based methods, on the other hand, require strong assumptions for global convergence, and are often conceptual methods which can be implemented in a rather simplified form merely. Hence the exchange and the discretization methods are often used only for the first stage of the solution process, whereas the reduction-based methods are typically employed only in the final stage in order to provide a higher accuracy of the solution and a better rate of convergence. Wu et al. [33] proposed an iterative method for solving KKT systems of the SIP in which they drop some redundant points at certain iterations. Qi et al. [27] and Li et al. [20] presented semismooth/smoothing Newton methods in which the index set Ω is specified by $\Omega = \{s \mid c_j(s) \leq 0 \ (j = 1, \dots, r)\}$ with twice continuously differentiable functions $c_j : \mathbb{R}^m \rightarrow \mathbb{R} \ (j = 1, \dots, r)$. They proved that their algorithms have nice convergence properties, but those two methods cannot ensure the feasibility. For linear SIPs, Lai and Wu [19] proposed the explicit algorithm in which they solve a linear program with a finite index set E_k . They dropped out redundant points in E_k at each iteration and only kept active points ensuring $|E_k| \leq n$ for each k , and hence the algorithm is very efficient in saving the computational time.

In this paper, we focus on the following convex semi-infinite program with “second-order cone constraints” (SOCCSIP):

$$\begin{aligned}
 \text{(SOCCSIP)} \quad & \text{minimize } f(x) \\
 & \text{subject to } x \in \mathcal{K}, \quad c(x, s) \leq 0 \quad \forall s \in \Omega,
 \end{aligned} \tag{1.1}$$

where $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is continuously differentiable and convex, $c : \mathbb{R}^{n+m} \rightarrow \mathbb{R}$ is continuously differentiable and convex with respect to x , $\Omega \subseteq \mathbb{R}^m$ is a given compact set playing a role of index set, and \mathcal{K} is the Cartesian product of several second order cones (SOCs), i.e., $\mathcal{K} = \mathcal{K}^{n_1} \times \dots \times \mathcal{K}^{n_m}$ with $n = n_1 + \dots + n_m$ and

$$\mathcal{K}^{n_j} := \begin{cases} \left\{ x = (x_1, \tilde{x}) \in \mathbb{R} \times \mathbb{R}^{n_j-1} \mid x_1 \geq \|\tilde{x}\|_2 \right\} & (n_j \geq 2) \\ \{x \in \mathbb{R} \mid x \geq 0\} & (n_j = 1). \end{cases}$$

In the remainder of the paper, the Euclidean norm $\|\cdot\|_2$ is denoted as $\|\cdot\|$.

The SOCCSIP can be applied to many practical problems. One possible application is a robust optimization problem [2–4, 26] with spherical or ellipsoidal uncertainty. Since the typical example will be given in Sect. 5, we omit the details here. Another application is the finite impulse response (FIR) filter design problem. Let $h_1, h_2, \dots, h_{2N} \in \mathbb{R}$ be the

coefficients (impulse response) of a FIR filter with length $2N$. Then the frequency response function $H : [0, 2\pi] \rightarrow \mathbb{C}$ is defined as $H(\omega) := \sum_{k=1}^{2N} h_k e^{-j(k-1)\omega}$, where $j := \sqrt{-1}$. Suppose that the filter coefficients have symmetric structure, i.e., $x_k := h_k = h_{2N+1-k}$ for $k = 1, \dots, N$. According to the calculation steps in [22, Sect. 3.3], we obtain $H(\omega) = e^{-j(N-\frac{1}{2})\omega} \tau(\omega)^\top x$ with $x := (x_1, x_2, \dots, x_N)^\top$ and $\tau(\omega) := 2(\cos(N - \frac{1}{2})\omega, \cos(N - \frac{3}{2})\omega, \dots, \cos \frac{1}{2}\omega)^\top$. We also have $|H(\omega)| = |\tau(\omega)^\top x|$ since $|e^{-j(N-\frac{1}{2})\omega}| = 1$. Now, we aim to design the filter as follows: (i) we let the filter magnitude $|H(\omega)|$ be upper-bounded by $\beta > 0$ in the stopband $\omega \in [\omega_s, \pi]$; (ii) we let $|H(\omega)|$ be as close as possible to 1 in logarithmic scale in the passband $\omega \in [0, \omega_p]$. Then, the FIR filter design problem is formulated as

$$\begin{aligned} &\text{Minimize}_x \quad \max_{\omega \in [0, \omega_p]} \left| \log |\tau(\omega)^\top x| - \log 1 \right| \\ &\text{subject to} \quad |\tau(\omega)^\top x| \leq \beta \quad (\forall \omega \in [\omega_s, \pi]), \end{aligned}$$

which can be reformulated equivalently as

$$\begin{aligned} &\text{Minimize}_{u,v,x,y} \quad u \\ &\text{subject to} \quad u \geq 0, v \geq 0, y \in \mathcal{K}^3, \\ &\quad (u + v, u - v, 2)^\top - y = 0, \\ &\quad v \leq \tau(\omega)^\top x \leq u \quad (\forall \omega \in [0, \omega_p]), \\ &\quad -\beta \leq \tau(\omega)^\top x \leq \beta \quad (\forall \omega \in [\omega_s, \pi]), \end{aligned}$$

by introducing the auxiliary variables $u, v \in \mathbb{R}$ and $y \in \mathbb{R}^3$. (More detailed reformulation techniques are found in [22]). This problem is certainly of the form SOCCSIP (1.1).

Since $x \in \mathcal{K}^n \Leftrightarrow u^\top x \geq 0 \quad (\forall u \in U)$ with $U = \{u \in \mathbb{R}^n \mid u_1 = \|(u_2, \dots, u_n)\| = 1\}$, the SOC constraint can be rewritten equivalently by an infinite number of linear constraints. Thus, SOCCSIP (1.1) can be cast as a standard convex SIP and existing SIP solvers can be applied. However, this approach is not sensible due to the following two aspects.

- (i) The dimension of the index set increases from m to $n + m$. (When $\mathcal{K} = \mathcal{K}^n$, the index set of the equivalent convex SIP becomes $\Omega \oplus U$).
- (ii) Some nice structures of the SOC such as the self-duality is destroyed, and hence the Euclidean Jordan algebra cannot be applied.

Indeed, the numerical results in [13] point out that the SOC constraint should not be handled as infinitely many linear constraints, but should be handled in a direct manner, especially when the dimension of the SOC is large.

The main goal of the paper is to design an explicit exchange method for solving SOCCSIP (1.1) and study its convergence properties. The algorithm inherits Lai and Wu’s explicit exchange technique [19], but we also introduce the relaxed scheme in which a maximization problem with respect to $s \in \Omega$ need not be solved in each iteration. (It suffices to find some $s \in \Omega$ such that a certain criterion with small $\eta > 0$ is satisfied. This scheme is also installed in the algorithms proposed in [13, 37]). When either of f or $c(\cdot, s)$ is strictly convex, we can obtain a finite convergence result without using the special structure of SOCs (Sect. 4.2). Therefore, even when both of f and $c(\cdot, s)$ are non-strictly convex, we can have the “approximate optimum” by solving the perturbed SOCCSIP in which the objective function is replaced by $f(x) + \varepsilon \|x\|^2$ with a very small $\varepsilon > 0$ (Sect. 4.4). However, without using such a regularization technique, we can have the finite convergence result when $c(\cdot, s)$ is affine or quadratic (Sect. 4.3). In this case, we have to utilize the SOC complementarity conditions and the spectral factorization techniques associated with Euclidean Jordan algebra. In

the classical studies on the cutting plane or the exchange type methods (see [18, 21, 23, 28, 32] and references therein), the convergence properties were analyzed in a componentwise manner. However, such analyses does not make sense for SOCCSIP anymore since the SOC does not have componentwisely independent structure.

Also, we have to mention the related works [13, 25, 37] published recently. The topics and the ideas studied in those papers are similar to our work in some parts. However, they are different in other parts, which are summarized as follows.

- When f and $c(\cdot, s)$ are affine, SOCCSIP (1.1) reduces to the SOCLSIP studied in [13]. However, the results for SOCLSIP cannot be applied to SOCCSIP (1.1) directly since, in case of SOCCSIP, we have to evaluate the residual values $f(x^{k+1}) - f(x^k) - \nabla_x f(x^k)^\top (x^{k+1} - x^k)$ and $c(x^{k+1}, s) - c(x^k, s) - \nabla_x c(x^k, s)^\top (x^{k+1} - x^k)$ in each iteration, which are absent when f and $c(\cdot, s)$ are affine.
- In [25], the authors study the regularized explicit exchange method for solving SIP with infinitely many conic constraints. However, they analyze the convergence property of the algorithm only for the problems whose constraint function is affine. (See Sects. 3 and 4 in [25]). On the other hand, we consider more general cases. Moreover, the regularization technique is not installed in our algorithm when $c(\cdot, s)$ is affine or quadratic.
- In [37], the authors propose the explicit exchange method, in which they introduce the finite index set $\Omega_0 \subset \Omega$ whose elements are never dropped throughout the iterations. We also use the same technique in the algorithm. However, [37] only focuses on the case where either f or $c(\cdot, s)$ is strictly convex, whereas we study the case where both f and $c(\cdot, s)$ are non-strictly convex.

This paper is organized as follows. In Sect. 2, we give some preliminaries needed for the later analyses. In Sect. 3, we develop an explicit exchange method for solving SOCCSIP (1.1). In Sect. 4, we establish the convergence analysis of the proposed algorithm. In Sect. 5, we give some numerical results.

2 Preliminaries

In this section, we give some fundamental knowledge on SOCCSIP (1.1) and the SOC which will be necessary in the subsequent sections.

Throughout the paper, we suppose that SOCCSIP (1.1) satisfies the following assumption.

Assumption A (i) The optimum set of SOCCSIP (1.1) is nonempty and compact. (ii) It holds $\inf\{f(x) \mid x \in \mathcal{K}\} < \inf\{f(x) \mid x \in \mathcal{K}, c(x, s) \leq 0 \ (\forall s \in \Omega)\}$. (iii) There exists an $\bar{x} \in \mathcal{K}$ such that $c(\bar{x}, s) < 0$ for all $s \in \Omega$.

This assumption is expected to hold in most cases. Assumption A (i) obviously holds when the optimum set is a singleton. Assumption A (ii) claims that there exists at least one index $\bar{s} \in \Omega$ such that the inequality constraint $c(x, \bar{s}) \leq 0$ is strictly active at the optimum. Therefore, if this assumption does not hold, then the infinite constraint $c(x, s) \leq 0 \ (\forall s \in \Omega)$ in SOCCSIP (1.1) does not make sense anymore. Assumption A (iii) is nothing more than the well-known Slater condition. Under this assumption we have the following theorem, which can be proved by using [5, Proposition 2.3.1], [1, Lemma 3.1], and the convergence theorem for the discretization method [28, Chap. 7].

Theorem 2.1 *Suppose that Assumptions A (i) and A (ii) hold. Then, there exists $\Omega_0 = \{s_1^0, \dots, s_{m_0}^0\} \subset \Omega$ such that the following statements hold:*

- f is level-bounded on the set $\mathcal{K} \cap \{x \mid c(x, s_i^0) \leq 0 \ (i = 1, 2, \dots, m_0)\}$.
- $\inf \{f(x) \mid x \in \mathcal{K}\} < \inf \{f(x) \mid x \in \mathcal{K}, c(x, s_i^0) \leq 0 \ (i = 1, 2, \dots, m_0)\}$.

We next give some properties on the SOC. We introduce the spectral factorization for a single SOC \mathcal{K}^ℓ in Euclidean Jordan algebra [9, 10]. For any vector $y := (y_1, \tilde{y}) \in \mathbb{R} \times \mathbb{R}^{\ell-1}$ with $\ell \geq 2$, its spectral factorization is defined by

$$y = \lambda_1(y)v_1(y) + \lambda_2(y)v_2(y),$$

where $\lambda_i(y) \in \mathbb{R}$ and $v_i(y) \in \mathbb{R}^\ell \ (i = 1, 2)$ are the spectral values and vectors, respectively, defined by

$$\begin{aligned} \lambda_i(y) &:= y_1 + (-1)^i \|\tilde{y}\|, \\ v_i(y) &:= \begin{cases} \frac{1}{2} \begin{pmatrix} 1, (-1)^i \frac{\tilde{y}}{\|\tilde{y}\|} \end{pmatrix} & \text{if } \tilde{y} \neq 0, \\ \frac{1}{2} (1, (-1)^i w) & \text{if } \tilde{y} = 0. \end{cases} \end{aligned} \tag{2.1}$$

Here, $w \in \mathbb{R}^{\ell-1}$ is an arbitrary vector with $\|w\| = 1$. It is obvious that

$$\begin{aligned} \lambda_1(y) \leq \lambda_2(y), \quad \lambda_1(y) \geq 0 &\Leftrightarrow y \in \mathcal{K}^\ell, \\ \lambda_1(y) = 0 &\Leftrightarrow y \in \text{bd } \mathcal{K}^\ell, \quad \lambda_1(y) > 0 \Leftrightarrow y \in \text{int } \mathcal{K}^\ell \end{aligned}$$

and

$$\|v_1(y)\| = \|v_2(y)\| = 1/\sqrt{2}, \quad v_1(y)^\top v_2(y) = 0.$$

Now, we study the relation between the complementarity on SOCs and the spectral factorization. The vectors $x \in \mathbb{R}^n$ and $z \in \mathbb{R}^n$ are said to satisfy the second-order cone complementarity if

$$x \in \mathcal{K}, \quad z \in \mathcal{K}, \quad x^\top z = 0. \tag{2.2}$$

It is easily seen that (2.2) holds if and only if

$$x_j \in \mathcal{K}^{n_j}, \quad z_j \in \mathcal{K}^{n_j}, \quad \text{and } x_j^\top z_j = 0, \quad (j = 1, \dots, m) \tag{2.3}$$

where x_j and z_j denote the Cartesian subvectors of x and z , respectively, i.e.,

$$x = (x_1, \dots, x_m), \quad z = (z_1, \dots, z_m) \in \mathbb{R}^{n_1} \times \dots \times \mathbb{R}^{n_m}. \tag{2.4}$$

Moreover, by [13, Prop. 2.3], we have

$$v_1(x_j) = v_2(z_j) \text{ and } v_2(x_j) = v_1(z_j).$$

3 The Explicit Exchange Method for SOCCSIP

In this section, we propose an explicit exchange method for solving SOCCSIP(1.1) and give some fundamental properties. In each iteration of the algorithm, we solve a finitely constrained convex program as a subproblem. For a finite set $E = \{s_1, \dots, s_m\} \subset \Omega$, let CSOCP(E) denote the finitely constrained convex second-order cone program defined as

$$\begin{aligned} \text{CSOCP}(E) : \quad & \text{minimize } f(x) \\ & \text{subject to } x \in \mathcal{K}, \quad c(x, s_j) \leq 0 \ (j = 1, \dots, m). \end{aligned}$$

Then, the first-order optimality condition of CSOCP(E) is given by

$$\begin{aligned}
 z &:= \nabla f(x) + \sum_{j=1}^m v(s_j) \nabla_x c(x, s_j) \in \mathcal{K}, \quad x \in \mathcal{K}, \quad x^\top z = 0, \\
 c(x, s_j) v(s_j) &= 0, \quad c(x, s_j) \leq 0, \quad v(s_j) \geq 0 \quad (j = 1, \dots, m),
 \end{aligned}
 \tag{3.1}$$

where $v(s_j)$ ($j = 1, \dots, m$) denotes the Lagrange multiplier corresponding to the constraint $c(x, s_j) \leq 0$. For more details on the optimality conditions, see e.g., [6, 7]. Actually, (3.1) reduces to a monotone second-order cone complementarity problem (SOCCP), which can be solved by existing algorithms [8, 10, 14, 15, 24]. Nonlinear second-order cone program such as CSOCP(E) is also studied by some researchers [17, 34], but they are still immature.

The concrete scheme of the algorithm is written as follows.

Algorithm 1

Step 0. Find a finite index set $\Omega_0 = \{s_1^0, \dots, s_{m_0}^0\}$ satisfying Theorem 2.1, and let E_0 be such that $\Omega_0 \subset E_0 \subset \Omega$ and $|E_0| < \infty$. Solve CSOCP(E_0) to obtain its optimum x^0 . Choose a small number $\eta > 0$ and set $k := 0$.

Step 1. Find an $s_{\text{new}}^k \in \Omega$ such that

$$c(x^k, s_{\text{new}}^k) > \eta.
 \tag{3.2}$$

If such an s_{new}^k does not exist, i.e., $\max_{s \in \Omega} c(x^k, s) \leq \eta$, then stop. Otherwise, let

$$\bar{E}_{k+1} := E_k \cup \{s_{\text{new}}^k\}.$$

Step 2. Solve CSOCP(\bar{E}_{k+1}) to obtain its optimum x^{k+1} and the Lagrange multipliers $\{v_{k+1}(s) \mid s \in \bar{E}_{k+1}\}$.

Step 3. Let

$$E_{k+1} := \Omega_0 \cup \{s \in \bar{E}_{k+1} \setminus \Omega_0 \mid v_{k+1}(s) > 0\}.$$

Let $k := k + 1$ and go to Step 1.

In Step 1, it is also possible to choose multiple elements satisfying (3.2). Although we merely deal with the single-point exchange scheme in the following analyses, they are also applicable to multiple exchange type algorithms. In Step 2, CSOCP(\bar{E}_{k+1}) can be solved by using an existing method. Step 3 is to remove the constraints that are inactive at the optimum x^{k+1} and corresponding to $t \in \bar{E}_{k+1} \setminus \Omega_0$. Here, we note that x^{k+1} also solves CSOCP(E_{k+1}). Moreover, the sequence $\{x^k\}$ is bounded since we have $\Omega_0 \subset \bar{E}_{k+1}$ for all k . (See Proposition 4.1 given later).

Next, we define some notations for convenience. For a finite set $E = \{s_1, \dots, s_m\} \subset \Omega$, we denote the feasible set and the optimal value of CSOCP(E) by $\mathcal{F}(E) \subset \mathbb{R}^n$ and $V(E) \in [-\infty, +\infty)$, respectively, i.e.,

$$\begin{aligned}
 \mathcal{F}(E) &:= \{x \mid x \in \mathcal{K}, \quad c(x, s_j) \leq 0 \quad (j = 1, \dots, m)\}, \\
 V(E) &:= \inf \{f(x) \mid x \in \mathcal{K}, \quad c(x, s_j) \leq 0 \quad (j = 1, \dots, m)\}.
 \end{aligned}$$

Moreover, we denote the optimal value of SOCCSIP (1.1) by $V^* \in \mathbb{R}$, i.e.,

$$V^* := \inf \{f(x) \mid x \in \mathcal{K}, \quad c(x, s) \leq 0 \quad (\forall s \in \Omega)\}.
 \tag{3.3}$$

Let $\{x^k\}$ and $\{v_k(s)\}$ be the sequences generated by Algorithm 1, and $\{z^k\}$ be defined as

$$z^k := \nabla f(x^k) + \sum_{s \in E_k} v_k(s) \nabla_x c(x^k, s). \tag{3.4}$$

Then, the complementarity slackness condition (3.1) yields

$$\begin{aligned} x^k \in \mathcal{K}, \quad z^k \in \mathcal{K}, \quad (x^k)^\top z^k &= 0, \\ c(x^k, s) \leq 0, \quad v_k(s) \geq 0, \quad c(x^k, s)v_k(s) &= 0, \quad \forall s \in E_k. \end{aligned} \tag{3.5}$$

Moreover, by using the spectral factorization for Cartesian subvectors x_j^k and z_j^k (see (2.4)), we define $\hat{x}_{ij}^k, \hat{z}_{ij}^k \in \mathbb{R}$ and $\hat{e}_{ij}^k \in \mathbb{R}^{n_j}$ for each k as follows:

- When $n_j = 1$,

$$\hat{x}_{1j}^k := \hat{x}_{2j}^k := \frac{1}{2}x_j^k, \quad \hat{z}_{1j}^k := \hat{z}_{2j}^k := \frac{1}{2}z_j^k, \quad \hat{e}_{1j}^k := \hat{e}_{2j}^k := 1. \tag{3.6}$$

- When $\left[n_j \geq 2 \text{ and } k = 1 \right]$ or $\left[n_j \geq 2, k \geq 2, \text{ and } \|\sqrt{2}v_1(x_j^k) - \hat{e}_{1j}^{k-1}\| \leq \|\sqrt{2}v_1(x_j^k) - \hat{e}_{2j}^{k-1}\| \right]$,

$$\begin{aligned} \hat{x}_{1j}^k &:= \lambda_1(x_j^k)/\sqrt{2}, \quad \hat{x}_{2j}^k := \lambda_2(x_j^k)/\sqrt{2}, \quad \hat{z}_{1j}^k := \lambda_2(z_j^k)/\sqrt{2}, \quad \hat{z}_{2j}^k := \lambda_1(z_j^k)/\sqrt{2}, \\ \hat{e}_{1j}^k &:= \sqrt{2}v_1(x_j^k) = \sqrt{2}v_2(z_j^k), \quad \hat{e}_{2j}^k := \sqrt{2}v_2(x_j^k) = \sqrt{2}v_1(z_j^k). \end{aligned} \tag{3.7}$$

- When $n_j \geq 2, k \geq 2$, and $\|\sqrt{2}v_1(x_j^k) - \hat{e}_{1j}^{k-1}\| > \|\sqrt{2}v_1(x_j^k) - \hat{e}_{2j}^{k-1}\|$,

$$\begin{aligned} \hat{x}_{1j}^k &:= \lambda_2(x_j^k)/\sqrt{2}, \quad \hat{x}_{2j}^k := \lambda_1(x_j^k)/\sqrt{2}, \quad \hat{z}_{1j}^k := \lambda_1(z_j^k)/\sqrt{2}, \quad \hat{z}_{2j}^k := \lambda_2(z_j^k)/\sqrt{2}, \\ \hat{e}_{1j}^k &:= \sqrt{2}v_2(x_j^k) = \sqrt{2}v_1(z_j^k), \quad \hat{e}_{2j}^k := \sqrt{2}v_1(x_j^k) = \sqrt{2}v_2(z_j^k). \end{aligned} \tag{3.8}$$

Then, we have

$$x^k = (x_j^k)_{j=1}^m = \left(\hat{x}_{1j}^k \hat{e}_{1j}^k + \hat{x}_{2j}^k \hat{e}_{2j}^k \right)_{j=1}^m, \quad z^k = (z_j^k)_{j=1}^m = \left(\hat{z}_{1j}^k \hat{e}_{1j}^k + \hat{z}_{2j}^k \hat{e}_{2j}^k \right)_{j=1}^m, \tag{3.9}$$

for each k . For such factorizations, we have the following proposition

Proposition 3.1 [13, Prop. 3.4] *Let x_j^k and z_j^k be factorized as (3.9). Then, for each $i = 1, 2, j = 1, \dots, m$, and $k \geq 1$, the following statements hold.*

- (a) $\max(\hat{x}_{ij}^k, \hat{z}_{ij}^k) \geq 0$ and $\min(\hat{x}_{ij}^k, \hat{z}_{ij}^k) = 0$.
- (b) (i) $\|\hat{e}_{ij}^k\| = 1$. (ii) $\hat{e}_{ij}^k \in \text{bd } \mathcal{K}^{n_j}$ and $(\hat{e}_{1j}^k)^\top \hat{e}_{2j}^k = 0$ if $n_j \geq 2$
- (c) $(\hat{e}_{ij}^k)^\top \hat{e}_{ij}^{k+1} \geq 1/2$.

We also define

$$\begin{aligned} d^k &:= x^{k+1} - x^k, \\ F_k &:= f(x^{k+1}) - f(x^k) - \nabla f(x^k)^\top d^k, \\ G_k &:= f(x^k) - f(x^{k+1}) + \nabla f(x^{k+1})^\top d^k \\ P_k(s) &:= c(x^{k+1}, s) - c(x^k, s) - \nabla_x c(x^k, s)^\top d^k, \\ Q_k(s) &:= c(x^k, s) - c(x^{k+1}, s) + \nabla_x c(x^{k+1}, s)^\top d^k. \end{aligned}$$

Since f and $c(\cdot, s)$ are continuously differentiable and convex, we have

$$F_k = o(\|d^k\|), \quad G_k = o(\|d^k\|), \quad F_k \geq 0, \quad G_k \geq 0$$

$$P_k(s) = o(\|d^k\|), \quad Q_k = o(\|d^k\|), \quad P_k(s) \geq 0, \quad Q_k(s) \geq 0.$$

4 Convergence Analysis

In this section, we show that the proposed algorithm terminates in a finite number of iterations under some mild conditions. Furthermore, we prove that the last output is sufficiently close to the optimal solution of SOCCSIP (1.1) if the criterion value is sufficiently close to zero.

4.1 Some Technical Propositions

In this subsection, we give some technical propositions that are important and convenient for analyzing the convergence of Algorithm 1.

The following proposition shows the boundedness of the generated sequence. The proof can be obtained easily since Theorem 2.1 guarantees the compactness of $\{x \mid f(x) \leq V^*\} \cap \mathcal{F}(\Omega_0) \supset \{x^k\}$, where V^* is defined by (3.3).

Proposition 4.1 *The sequence $\{x^k\}$ generated by Algorithm 1 is bounded, i.e., there exists $M > 0$ such that $\|x^k\| \leq M$ for all k .*

The following proposition evaluates the increment of the optimal value of CSOCP(\bar{E}_k) in each iteration.

Proposition 4.2 *For all $k \geq 0$, we have*

$$f(x^{k+1}) - f(x^k) = (z^k)^\top x^{k+1} + F_k + \sum_{s \in E_k} v_k(s) \left(P_k(s) - c(x^{k+1}, s) \right) \tag{4.1}$$

$$= -(z^{k+1})^\top x^k - G_k - \sum_{s \in E_k} v_{k+1}(s) \left(Q_k(s) - c(x^k, s) \right)$$

$$+ v_{k+1}(s_{\text{new}}^k) \left(c(x^k, s_{\text{new}}^k) - Q_k(s_{\text{new}}^k) \right) \tag{4.2}$$

$$= -(z^{k+1})^\top x^k - G_k - \sum_{s \in E_k} v_{k+1}(s) \left(Q_k(s) - c(x^k, s) \right)$$

$$- v_{k+1}(s_{\text{new}}^k) \nabla_x c(x^{k+1}, s_{\text{new}}^k)^\top d^k \tag{4.3}$$

Proof By the definitions of d^k, z^k, F_k and $P_k(s)$, we have

$$(z^k)^\top x^{k+1} = (z^k)^\top \left(x^{k+1} - x^k \right) + \sum_{s \in E_k} v_k(s) c(x^k, s)$$

$$= \left(\nabla f(x^k) + \sum_{s \in E_k} v_k(s) \nabla_x c(x^k, s) \right)^\top d^k + \sum_{s \in E_k} v_k(s) c(x^k, s)$$

$$= \nabla f(x^k)^\top d^k + \sum_{s \in E_k} v_k(s) \left(\nabla_x c(x^k, s)^\top d^k + c(x^k, s) \right)$$

$$= f(x^{k+1}) - f(x^k) - F_k + \sum_{s \in E_k} v_k(s) \left(c(x^{k+1}, s) - P_k(s) \right),$$

where the first equality follows from (3.5). Thus we have (4.1).

Next, we show the last two equalities. By the definitions of d^k , z^{k+1} , G_k and $Q_k(s)$, we have

$$\begin{aligned} -(z^{k+1})^\top x^k &= (z^{k+1})^\top (x^{k+1} - x^k) + \sum_{s \in E_{k+1}} v_{k+1}(s) c(x^{k+1}, s) \\ &= \left(\nabla f(x^{k+1}) + \sum_{s \in E_{k+1}} v_{k+1}(s) \nabla_x c(x^{k+1}, s) \right)^\top d^k \\ &\quad + \sum_{s \in E_{k+1}} v_{k+1}(s) c(x^{k+1}, s) \\ &= \nabla f(x^{k+1})^\top d^k + \sum_{s \in E_{k+1}} v_{k+1}(s) \left(\nabla_x c(x^{k+1}, s)^\top d^k + c(x^{k+1}, s) \right) \\ &= \nabla f(x^{k+1})^\top d^k + \sum_{s \in E_{k+1}} v_{k+1}(s) \left(Q_k(s) - c(x^k, s) \right) \\ &= \nabla f(x^{k+1})^\top d^k + \sum_{s \in E_k} v_{k+1}(s) \left(Q_k(s) - c(x^k, s) \right) \\ &\quad + v_{k+1}(s_{\text{new}}^k) \left(Q_k(s_{\text{new}}^k) - c(x^k, s_{\text{new}}^k) \right) \\ &= \nabla f(x^{k+1})^\top d^k + \sum_{s \in E_k} v_{k+1}(s) \left(Q_k(s) - c(x^k, s) \right) \\ &\quad + v_{k+1}(s_{\text{new}}^k) \nabla_x c(x^{k+1}, s_{\text{new}}^k)^\top d^k, \end{aligned}$$

where the first equality follows from (3.5) with $k := k + 1$, the fifth equality follows from $\bar{E}_{k+1} = E_k \cup \{s_{\text{new}}^k\}$ and $v_{k+1}(s) = 0$ for any $s \in \bar{E}_{k+1} \setminus E_{k+1}$, and the last equality holds since $v_{k+1}(s_{\text{new}}^k) c(x^{k+1}, s_{\text{new}}^k) = 0$. Thus we have (4.2) and (4.3). \square

Note that we have $F_k \geq 0$, $v_k(s) \geq 0$, $c(x^{k+1}, s) \leq 0$, and $P_k(s) \geq 0$ for any $s \in E_k$. Moreover, $z^k \in \mathcal{K}$ and $x^{k+1} \in \mathcal{K}$ entail $(z^k)^\top x^{k+1} \geq 0$. These inequalities and (4.1) lead us to the following corollary.

Corollary 4.1 *The sequence of optimal values $\{f(x^k)\}$ of $\{\text{CSOCP}(E_k)\}$ is monotonically nondecreasing, i.e.,*

$$f(x^0) \leq f(x^1) \leq \dots \leq f(x^k) \leq f(x^{k+1}) \leq \dots \leq V^*.$$

The following proposition shows that the distance between x^k and x^{k+1} never converges to 0.

Proposition 4.3 *Let $\{x^k\}$ be the sequence generated by Algorithm 1. Then, there exists $d_{\min} > 0$ such that $\|x^k - x^{k+1}\| \geq d_{\min}$ for all k .*

Proof Note that function $c(\cdot, s_{\text{new}}^k)$ is locally Lipschitzian since it is convex and $c(x, s_{\text{new}}^k) < \infty$ for any $x \in \mathbb{R}^n$. Also, we have $c(x^k, s_{\text{new}}^k) > \eta$ and $c(x^{k+1}, s_{\text{new}}^k) \leq 0$ for all k . Thus, noticing the boundedness of $\{x^k\}$ and Ω , we have

$$\begin{aligned} 0 < \eta &\leq c\left(x^k, s_{\text{new}}^k\right) - c\left(x^{k+1}, s_{\text{new}}^k\right) \\ &\leq L \|x^k - x^{k+1}\|, \end{aligned}$$

where $L > 0$ is the Lipschitzian constant. Hence, letting $d_{\min} := \eta/L > 0$, we obtain the result. \square

The following two propositions show that the sequences $\{\sum_{s \in E_k} v_k(s)\}$ and $\{\|z^k\|\}$ are bounded, and $\sum_{s \in E_k} v_k(s)$ is positive away from 0.

Proposition 4.4 *Let $\{x^k\}$ and $\{v_k\}$ be the sequences generated by Algorithm 1, and $\{z^k\}$ be defined by (3.4). Then, there exists $M > 0$ such that $\|z^k\| \leq M$ and $\sum_{s \in E_k} v_k(s) \leq M$ for all k .*

Proof Let $\bar{x} \in \mathcal{K}$ and $\beta > 0$ be chosen so that $c(\bar{x}, s) \leq -\beta$ for any $s \in \Omega$. (Such a vector and a positive number exist from Assumption A). Then, we have

$$\begin{aligned} 0 &\leq (z^k)^\top (\bar{x} - x^k) + \sum_{s \in E_k} v_k(s) c(x^k, s) \\ &= \nabla f(x^k)^\top (\bar{x} - x^k) + \sum_{s \in E_k} v_k(s) \left(\nabla_x c(x^k, s)^\top (\bar{x} - x^k) + c(x^k, s) \right) \\ &\leq \nabla f(x^k)^\top (\bar{x} - x^k) + \sum_{s \in E_k} v_k(s) c(\bar{x}, s) \\ &\leq \nabla f(x^k)^\top (\bar{x} - x^k) - \beta \sum_{s \in E_k} v_k(s), \end{aligned}$$

where the first inequality follows from $(z^k)^\top \bar{x} \geq 0$ and (3.5), the second inequality is due to the convexity of $c(\cdot, s)$, and the last inequality follows from $c(\bar{x}, s) \leq -\beta$. Hence, we have $\sum_{s \in E_k} v_k(s) \leq \nabla f(x^k)^\top (\bar{x} - x^k) / \beta$, which implies the boundedness of $\{\sum_{s \in E_k} v_k(s)\}$ since ∇f is continuous and $\{x^k\}$ is bounded. The boundedness of $\{\|z^k\|\}$ can be easily shown by the definition of z^k and the boundedness of $\{\sum_{s \in E_k} v_k(s)\}$. \square

Proposition 4.5 *Let $\{x^k\}$ and $\{v_k\}$ be the sequences generated by Algorithm 1. Then, there exists $\alpha > 0$ such that $\sum_{s \in E_k} v_k(s) \geq \alpha$ for all k .*

Proof Assume that the statement does not hold for contradiction. Then, Algorithm 1 does not terminate finitely, and $\liminf_{k \rightarrow \infty} \sum_{s \in E_k} v_k(s) = 0$. Since $\{x^k\}$ is bounded from Proposition 4.1, there exists $K \subseteq \{1, 2, \dots\}$ such that $\bar{x} := \lim_{k \rightarrow \infty, k \in K} x^k$ and $\lim_{k \rightarrow \infty, k \in K} \sum_{s \in E_k} v_k(s) = 0$. Also, we have $\lim_{k \rightarrow \infty, k \in K} z^k = \nabla f(\bar{x})$ from (3.4) and the boundedness of $\{\nabla_x c(x^k, s)\}$. Thus, by (3.5), we have $\bar{x} \in \mathcal{K}$, $\nabla f(\bar{x}) \in \mathcal{K}$ and $\bar{x}^\top \nabla f(\bar{x}) = 0$, which imply $f(\bar{x}) = \min\{f(x) \mid x \in \mathcal{K}\}$.¹ Hence, by Theorem 2.1 and $\Omega_0 \subset \bar{E}_k$, we must have $f(\bar{x}) < V(\Omega_0) \leq V(\bar{E}_k) = f(x^k)$. However, this contradicts $f(x^k) \leq f(x^{k+1}) \leq \dots \leq f(\bar{x})$. \square

4.2 Finite Termination for Strictly Convex Case

When f or $c(\cdot, s)$ is strictly convex, we can readily obtain the finite termination as follows.

Assumption B At least one of the following statements holds: (a) f is strictly convex; (b) $c(\cdot, s)$ is strictly convex for any $s \in \Omega$.

Theorem 4.1 *Suppose that Assumption B holds. Then, Algorithm 1 terminates in a finite number of iterations.*

Proof We can prove the theorem by using Corollary 4.1 and the argument similar to the proof of [37, Thm. 3.1]. (Here, we do not use the special structure of the SOC). \square

¹ Function f and cone \mathcal{K} are convex. Moreover, the KKT conditions of the problem “ $\min f(x)$ s.t. $x \in \mathcal{K}$ ” can be written as $x \in \mathcal{K}$, $\nabla f(x) \in \mathcal{K}$ and $x^\top \nabla f(x) = 0$.

4.3 Finite Termination Without Strict Convexity

In this section, we discuss the finite termination property of Algorithm 1 without assuming the strict convexity on f or $c(\cdot, s)$. We first consider the case (Case 1) where the constraint function $c(\cdot, s)$ is affine or quadratic. Then, we consider the more general case (Case 2).

Case 1: Affine or Quadratic Constraint Function

Now, we first suppose that the following assumption holds.

Assumption C The constraint function $c(\cdot, s)$ satisfies either of the following two conditions:

- (a) $c(\cdot, s)$ is affine for any $s \in \Omega$;
- (b) c is quadratic with respect to x , i.e., it is given as

$$c(x, s) := x^\top M(s)x + 2q(s)^\top x + r(s),$$

where $M : \Omega \rightarrow \mathbb{R}^{n \times n}$, $q : \Omega \rightarrow \mathbb{R}^n$ and $r : \Omega \rightarrow \mathbb{R}$ are continuous, and $M(s) \in \mathbb{R}^{n \times n}$ is symmetric and positive semidefinite for any $s \in \Omega$. Moreover, it holds that $q(s)^\top \xi \neq 0$ for any $s \in \Omega$ and $\xi \in \mathbb{R}^n \setminus \{0\}$ with $\xi^\top M(s)\xi = 0$.

Then, we provide two lemmas below, which play crucial roles in the finite convergence analysis appearing later.

Lemma 4.1 *Suppose that either of Assumption C(a) or C(b) holds. Then, there exists a $\mu > 0$ such that, for all k ,*

$$\nabla_x c(x^{k+1}, s_{\text{new}}^k)^\top (x^k - x^{k+1}) \geq \mu. \tag{4.4}$$

Proof When Assumption C(a) holds, we have $\nabla_x c(x^{k+1}, s_{\text{new}}^k)^\top (x^k - x^{k+1}) = c(x^k, s_{\text{new}}^k) - c(x^{k+1}, s_{\text{new}}^k)$ since $c(\cdot, s_{\text{new}}^k)$ is affine. Moreover, we have $c(x^k, s_{\text{new}}^k) > \eta$ and $c(x^{k+1}, s_{\text{new}}^k) \leq 0$ by Algorithm 1. Therefore, by letting $\mu := \eta$, we readily have (4.4).

Next we consider the case where Assumption C(b) holds. Let $s^k \in E_k$ be an arbitrary element with $v_k(s^k) > 0$. Let $\xi^k \in \mathbb{R}^n$ be any vector such that $\|\xi^k\| = 1$ and $\nabla_x c(x^k, s^k)^\top \xi^k = 2(M(s^k)x^k + q(s^k))^\top \xi^k = 0$. Then we have

$$\begin{aligned} c(x^k + \xi^k, s^k) &= (x^k + \xi^k)^\top M(s^k)(x^k + \xi^k) + 2q(s^k)^\top (x^k + \xi^k) + r(s^k) \\ &= \left[(x^k)^\top M(s^k)x^k + 2q(s^k)^\top x^k + r(s^k) \right] \\ &\quad + (\xi^k)^\top M(s^k)\xi^k + 2(M(s^k)x^k + q(s^k))^\top \xi^k \\ &= (\xi^k)^\top M(s^k)\xi^k, \end{aligned}$$

where the last equality follows since $\nabla_x c(x^k, s^k)^\top \xi^k = 0$ and $c(x^k, s^k) = 0$ from $v_k(s^k) > 0$.

We first show that there exists $\delta' > 0$ such that

$$(\xi^k)^\top M(s^k)\xi^k \geq \delta' \tag{4.5}$$

for all k . Suppose for contradiction that there does not exist such an $\delta' > 0$. Then, we must have $\liminf_{k \rightarrow \infty} (\xi^k)^\top M(s^k)\xi^k = 0$. Since $\{x^k\}$, $\{\xi^k\}$ and $\{s^k\}$ are bounded and $M(s)$ is continuous, there exists $K \subset \{0, 1, \dots\}$ such that $\lim_{k \rightarrow \infty, k \in K} x^k = \bar{x}$, $\lim_{k \rightarrow \infty, k \in K} \xi^k = \bar{\xi}$, $\lim_{k \rightarrow \infty, k \in K} s^k = \bar{s}$, and $\bar{\xi}^\top M(\bar{s})\bar{\xi} = 0$. Notice that $\bar{\xi}^\top M(\bar{s})\bar{\xi} = 0$ implies $M(\bar{s})\bar{\xi} = 0$. Moreover, we have $\nabla_x c(x, \bar{s})^\top \bar{\xi} = 0$ since $\nabla_x c(x^k, s^k)^\top \xi^k = 0$ for all k . We thus have $0 = \nabla_x c(\bar{x}, \bar{s})^\top \bar{\xi} = 2(M(\bar{s})\bar{x} + q(\bar{s}))^\top \bar{\xi} = q(\bar{s})^\top \bar{\xi}$. However, this contradicts the assumption.

Next, we show that there exists $\delta'' > 0$ such that

$$-\nabla_x c(x^k, s^k)^\top (x^{k+1} - x^k) \geq \delta'' \tag{4.6}$$

for all k . Let $\mathcal{P}_k := \{x \mid \nabla_x c(x^k, s^k)^\top (x - x^k)\} = 0, \tilde{x}^{k+1}$ be the Euclidean projection of x^{k+1} onto \mathcal{P}_k , and θ_k be the angle between $x^{k+1} - x^k$ and $\tilde{x}^{k+1} - x^k$. Notice that we always have $0 \leq \theta_k \leq \pi/2$ since $(x^{k+1} - x^k)^\top (\tilde{x}^{k+1} - x^k) = ((\tilde{x}^{k+1} - x^k) - (x^{k+1} - x^k))^\top (\tilde{x}^{k+1} - x^k) = \|\tilde{x}^{k+1} - x^k\|^2$, where the last equality follows from $(\tilde{x}^{k+1} - x^k) \perp (x^{k+1} - \tilde{x}^{k+1})$. If $\theta_k > \pi/4$, then we have $-\nabla_x c(x^k, s^k)^\top (x^{k+1} - x^k) = \|\nabla_x c(x^k, s^k)\| \|x^{k+1} - x^k\| \cos(\pi/2 - \theta_k) > \gamma d_{\min}/\sqrt{2}$, where $d_{\min} > 0$ is defined in Proposition 4.3 and $\gamma > 0$ is the positive number such that

$$\|\nabla_x c(x^k, s^k)\| \geq \gamma \tag{4.7}$$

for all k .² If $\theta_k \leq \pi/4$, then we have $\|\tilde{x}^{k+1} - x^k\| = \|x^{k+1} - x^k\| \cos \theta_k \geq d_{\min}/\sqrt{2}$. Hence, letting $\xi^k := (\tilde{x}^{k+1} - x^k)/\|\tilde{x}^{k+1} - x^k\|$ and $\tau_k := \|\tilde{x}^{k+1} - x^k\|$, we have

$$\begin{aligned} c(\tilde{x}^{k+1}, s^k) &= (\tilde{x}^{k+1})^\top M(s^k)\tilde{x}^{k+1} + 2q(s^k)^\top \tilde{x}^{k+1} + r(s^k) \\ &= (x^k + \tau_k \xi^k)^\top M(s^k)(x^k + \tau_k \xi^k) + 2q(s^k)^\top (x^k + \tau_k \xi^k) + r(s^k) \\ &= \tau_k^2 (\xi^k)^\top M(s^k) \xi^k \\ &\geq d_{\min}^2 \delta' / 2, \end{aligned}$$

where the last equality follows since $c(x^k, s^k) = 0$ and $\nabla_x c(x^k, s^k)^\top \xi^k = 0$, and the inequality is due to (4.5) and $\tau_k = \|\tilde{x}^{k+1} - x^k\| \geq d_{\min}/\sqrt{2}$. Since $c(x^{k+1}, s^k) \leq 0$ and c is locally Lipschitzian, there exists $L > 0$ such that $d_{\min}^2 \delta' / 2 \leq c(\tilde{x}^{k+1}, s^k) - c(x^{k+1}, s^k) \leq L \|\tilde{x}^{k+1} - x^{k+1}\|$, that is, $\|\tilde{x}^{k+1} - x^{k+1}\| \geq d_{\min}^2 \delta' / (2L)$. So we have

$$-\nabla_x c(x^k, s^k)^\top (x^{k+1} - x^k) = \|\nabla_x c(x^k, s^k)\| \|\tilde{x}^{k+1} - x^{k+1}\| \geq \gamma d_{\min}^2 \delta' / (2L),$$

where $\gamma > 0$ is a positive number given by (4.7).

Finally, we show that (4.4) holds. Notice that

$$\begin{aligned} \nabla f(x^{k+1})^\top (x^{k+1} - x^k) &\geq \nabla f(x^k)^\top (x^{k+1} - x^k) \\ &= \left[z^k - \sum_{s \in E_k} v_k(s) \nabla_x c(x^k, s) \right]^\top (x^{k+1} - x^k) \\ &\geq \sum_{s \in E_k} v_k(s) \delta'' \geq \alpha \delta'', \end{aligned} \tag{4.8}$$

where the first inequality is due to the convexity of f , the second inequality follows from (4.6) and $(z^k)^\top x^k = 0 \leq (z^k)^\top x^{k+1}$, and the last inequality is due to Proposition 4.5. We thus have

² Assumption A(iii) and Proposition 4.1 yield $0 < -c(\bar{x}, \bar{s}_{\max}) \leq c(x^k, s^k) - c(\bar{x}, s^k) \leq \nabla_x c(x^k, s^k)^\top (x^k - \bar{x}) \leq \|\nabla_x c(x^k, s^k)\| (M + \|\bar{x}\|)$, where $\bar{s}_{\max} := \operatorname{argmax}_{s \in \Omega} c(\bar{x}, s)$. Therefore, we can choose $\gamma := -c(\bar{x}, \bar{s}_{\max}) / (M + \|\bar{x}\|) > 0$.

$$\begin{aligned}
 & \nu_{k+1} \left(s_{\text{new}}^k \right) \nabla_x c \left(x^{k+1}, s_{\text{new}}^k \right)^\top \left(x^k - x^{k+1} \right) \\
 &= -\nu_{k+1} \left(s_{\text{new}}^k \right) \nabla_x c \left(x^{k+1}, s_{\text{new}}^k \right)^\top \left(x^{k+1} - x^k \right) \\
 &= \left[-z^{k+1} + \nabla f(x^{k+1}) + \sum_{s \in E_k} \nu_{k+1}(s) \nabla_x c(x^{k+1}, s) \right]^\top (x^{k+1} - x^k) \\
 &\geq \alpha \delta'' + \sum_{s \in E_k} \nu_{k+1}(s) \left[Q_k(s) - c(x^k, s) + c(x^{k+1}, s) \right] \\
 &\geq \alpha \delta'',
 \end{aligned} \tag{4.9}$$

where the first inequality is due to (4.8), $(z^{k+1})^\top x^{k+1} = 0 \leq (z^{k+1})^\top x^k$, and the definition of $Q_k(s)$, and the last inequality follows from $\nu_{k+1}(s) \geq 0$, $Q_k(s) \geq 0$, $c(x^k, s) \leq 0$ and $\nu_{k+1}(s)c(x^{k+1}, s) = 0$. By Proposition 4.4, we have $\nu_{k+1}(s_{\text{new}}^k) \leq M$. Hence, dividing both sides of (4.9) by $\nu_{k+1}(s_{\text{new}}^k) > 0$, we obtain

$$\nabla_x c \left(x^{k+1}, s_{\text{new}}^k \right)^\top \left(x^k - x^{k+1} \right) \geq \frac{\alpha \delta''}{\nu_{k+1}(s_{\text{new}}^k)} \geq \frac{\alpha \delta''}{M},$$

which implies (4.4) with $\mu := \alpha \delta'' / M$. □

Lemma 4.2 *Let $\theta : \mathbb{R}^n \rightarrow \mathbb{R}$ be an arbitrary continuously differentiable convex function, and $x, y \in \mathbb{R}^n$ be arbitrary vectors. Then we have*

$$\theta(y) - \theta(x) - \nabla \theta(x)^\top (y - x) = 0 \iff \theta(x) - \theta(y) + \nabla \theta(y)^\top (y - x) = 0.$$

Proof The above formula holds evidently when $x = y$. Also, if we have (\implies) , then (\impliedby) holds automatically by swapping x for y . Therefore, we only show (\implies) with $x \neq y$.

Let x and y be arbitrary vectors such that $x \neq y$ and $\theta(y) - \theta(x) - \nabla \theta(x)^\top (y - x) = 0$. Choose $\alpha \in (0, 1)$ arbitrarily. Then, we have

$$\begin{aligned}
 0 &= \alpha \theta(y) - \alpha \theta(x) - \alpha \nabla \theta(x)^\top (y - x) \\
 &= [(1 - \alpha)\theta(x) + \alpha \theta(y) - \theta((1 - \alpha)x + \alpha y)] \\
 &\quad + \left[\theta((1 - \alpha)x + \alpha y) - \theta(x) - \nabla \theta(x)^\top ((1 - \alpha)x + \alpha y - x) \right],
 \end{aligned}$$

which implies

$$(1 - \alpha)\theta(x) + \alpha \theta(y) - \theta((1 - \alpha)x + \alpha y) = 0 \tag{4.10}$$

and $\theta((1 - \alpha)x + \alpha y) - \theta(x) - \nabla \theta(x)^\top ((1 - \alpha)x + \alpha y - x) = 0$ since θ is convex. Hence, we have

$$\begin{aligned}
 -\nabla \theta(y)^\top (y - x) &= \nabla \theta(y)^\top (x - y) \\
 &= \lim_{t \downarrow 0} (\theta((1 - t)y + tx) - \theta(y)) / t \\
 &= \lim_{t \downarrow 0} ((1 - t)\theta(y) + t\theta(x) - \theta(y)) / t \\
 &= \theta(x) - \theta(y),
 \end{aligned}$$

where the third equality follows from (4.10) with $\alpha := 1 - t$. This completes the proof. □

To show the finite iteration of the algorithm, we further introduce the following assumption.

Assumption D (*strict complementarity and activeness*) There exists a small number $\delta > 0$ such that the following statements hold for each k :

- (i) $\max(\hat{x}_{ij}^k, \hat{z}_{ij}^k) \geq \delta$; (ii) $\max(v_k(s), -c(x^k, s)) \geq \delta \ (\forall s \in E_k)$; (iii) $v_{k+1}(s_{\text{new}}^k) \geq \delta$.

Statements (i) and (ii) claim that the *strict complementarity* should be satisfied in (3.5) in the sense of $\delta > 0$. Due to the complementarity, we always have $\min(v_k(s), -c(x^k, s)) = \min(\hat{x}_{ij}^k, \hat{z}_{ij}^k) = 0$. Statement (iii) implies that the newly added constraint is sufficiently active.

Now, by using the aforementioned assumption and lemma, we provide the theorem for the finite iteration of Algorithm 1.

Theorem 4.2 *Suppose that Assumptions C and D holds. Then, Algorithm 1 terminates in a finite number of iterations.*

Proof Suppose to the contrary that Algorithm 1 does not finitely terminate. Then, by Corollary 4.1 we have

$$f(x^1) \leq \dots \leq f(x^k) \leq f(x^{k+1}) \leq \dots \leq V^*,$$

which implies

$$\lim_{k \rightarrow \infty} (f(x^{k+1}) - f(x^k)) = 0. \tag{4.11}$$

Hence, each term in (4.1) also converges to 0 due to its nonnegativity. Let $E_k^+ := \{s \in E_k \mid v_k(s) > 0\} = \{s \in E_k \mid v_k(s) \geq \delta\}$, and $s_{\text{max}}^k := \operatorname{argmax}_{s \in E_k^+} Q_k(s)$. Moreover, noticing the boundedness of $\{x^k\}$ and Ω , let $(\bar{x}, \bar{x}_+, \bar{s}_{\text{max}}) \in \mathbb{R} \times \mathbb{R} \times \Omega$ be an arbitrary accumulation point of $\{(x^k, x^{k+1}, s_{\text{max}}^k)\}$. Then, there exists an index set $K \subseteq \{0, 1, 2, \dots\}$ such that $\lim_{k \rightarrow \infty, k \in K} (x^k, x^{k+1}, s_{\text{max}}^k) = (\bar{x}, \bar{x}_+, \bar{s}_{\text{max}})$. Since we have Proposition 4.3, it must hold $\bar{x} \neq \bar{x}_+$.

We first show that, for each $s \in \Omega_0$, there exists \bar{k} such that either

$$v_k(s) \geq \delta \ (\forall k \geq \bar{k}) \quad \text{or} \quad v_k(s) = 0 \ (\forall k \geq \bar{k}). \tag{4.12}$$

Fix $s \in \Omega_0$ arbitrarily. Then, by Assumption D (ii), we must have either $\limsup_{k \rightarrow \infty} v_k(s) = 0$ or $\limsup_{k \rightarrow \infty} v_k(s) \geq \delta$. If $\limsup_{k \rightarrow \infty} v_k(s) = 0$, then we obviously have $v_k(s) = 0$ for all k sufficiently large. If $\limsup_{k \rightarrow \infty} v_k(s) \geq \delta$, then there exists $K' \subset \{1, 2, \dots\}$ such that $|K'| = \infty$ and $v_k(s) \geq \delta$ for all $k \in K'$. Since $v_k(s)c(x^{k+1}, s)$ converges to 0, we have an $\varepsilon \in (0, \delta^2)$ and $\bar{k}' \geq \hat{k}$ such that $0 \leq -v_k(s)c(x^{k+1}, s) < \varepsilon$ for all $k \geq \bar{k}'$. Now, choose an arbitrary $\bar{k} \geq \bar{k}'$ such that $v_{\bar{k}}(s) \geq \delta$. Then, we have $0 \leq -c(x^{\bar{k}+1}, s) < \varepsilon/\delta < \delta$, which implies $c(x^{\bar{k}+1}, s) = 0$ and $v_{\bar{k}+1}(s) \geq \delta$ from Assumption D(ii). We thus obtain (4.12) recursively.

We next show

$$\lim_{k \rightarrow \infty} \sum_{s \in E_k} v_{k+1}(s)c(x^k, s) = 0, \quad \lim_{k \rightarrow \infty} G_k = 0, \quad \lim_{k \rightarrow \infty} \sum_{s \in E_k} v_{k+1}(s)Q_k(s) = 0. \tag{4.13}$$

We readily have $\lim_{k \rightarrow \infty} \sum_{s \in E_k} v_{k+1}(s)c(x^k, s) = 0$ since (4.12) implies that either $v_{k+1}(s)$ or $c(x^k, s)$ is 0 for all k sufficiently large. Since $\lim_{k \rightarrow \infty} F_k = 0$, we have

$$\lim_{k \rightarrow \infty, k \in K} F_k = f(\bar{x}_+) - f(\bar{x}) - \nabla f(\bar{x})^\top (\bar{x}_+ - \bar{x}) = 0.$$

Hence, by Lemma 4.2, we have

$$\lim_{k \rightarrow \infty, k \in K} G_k = f(\bar{x}) - f(\bar{x}_+) + \nabla f(\bar{x}_+)^\top (\bar{x}_+ - \bar{x}) = 0.$$

Since \bar{x} and \bar{x}_+ are arbitrary accumulation points, the above equality implies $\lim_{k \rightarrow \infty} G_k = 0$. Also, since $\lim_{k \rightarrow \infty} \sum_{s \in E_k} v_k(s) P_k(s) = 0$, $P_k(s) \geq 0$, $s_{\max}^k \in E_k^+$, and $v_k(s) \geq \delta$ for all $s \in E_k^+$, we have

$$0 = \lim_{k \rightarrow \infty, k \in K} P_k(s_{\max}^k) = c(\bar{x}_+, \bar{s}_{\max}) - c(\bar{x}, \bar{s}_{\max}) - \nabla_x c(\bar{x}, \bar{s}_{\max})^\top (\bar{x}_+ - \bar{x}).$$

Hence, by Lemma 4.2, we have $\lim_{k \rightarrow \infty, k \in K} Q_k(s_{\max}^k) = 0$. Now, by Proposition 4.4, there exists $M > 0$ such that $\sum_{s \in E_k^+} v_{k+1}(s) \leq \sum_{s \in \bar{E}_{k+1}} v_{k+1}(s) = \sum_{s \in E_{k+1}} v_{k+1}(s) \leq M$ for all k . Moreover, $v_{k+1}(s) = 0$ for all $s \in E_k \setminus E_k^+$ and sufficiently large k since we have (4.12) and $E_k \setminus E_k^+ \subset \Omega_0$. We thus have

$$0 \leq \sum_{s \in E_k} v_{k+1}(s) Q_k(s) = \sum_{s \in E_k^+} v_{k+1}(s) Q_k(s) \leq Q_k(s_{\max}^k) \sum_{s \in E_k^+} v_{k+1}(s) \leq M Q_k(s_{\max}^k),$$

which yields $\lim_{k \rightarrow \infty, k \in K} \sum_{s \in E_k} v_{k+1}(s) Q_k(s) = 0$. Since \bar{x} , \bar{x}_+ and \bar{s}_{\max} are arbitrary accumulation points, we have $\lim_{k \rightarrow \infty} \sum_{s \in E_k} v_{k+1}(s) Q_k(s) = 0$.

Now, choose a sufficiently small number $\varepsilon > 0$ arbitrarily. By Lemma 4.1, we have

$$-\nabla_x c(x^{k+1}, s_{\text{new}}^k)^\top d^k = \nabla_x c(x^{k+1}, s_{\text{new}}^k)^\top (x^k - x^{k+1}) \geq \mu > 0. \tag{4.14}$$

Hence, by (4.1) and (4.3) together with (4.11), (4.13) and (4.14), we have some positive integer $L = L(\varepsilon)$ such that

$$0 \leq (z^k)^\top x^{k+1} < \varepsilon, \quad (x^k)^\top z^{k+1} > \frac{\delta \mu}{2} =: \gamma \tag{4.15}$$

for all $k \geq L$. Choose $k \geq L$ arbitrarily, and let \mathcal{I}_1^k and \mathcal{I}_2^k be defined as

$$\mathcal{I}_1^k := \{(i, j) \mid \hat{x}_{ij}^k > 0\}, \quad \mathcal{I}_2^k := \{(i, j) \mid \hat{x}_{ij}^k = 0\}.$$

Then, we note that $\mathcal{I}_1^k \cup \mathcal{I}_2^k = \{1, 2\} \times \{1, \dots, m\}$, and

$$\begin{aligned} (i, j) \in \mathcal{I}_1^k &\iff \hat{x}_{ij}^k \geq \delta \iff \hat{z}_{ij}^k = 0, \\ (i, j) \in \mathcal{I}_2^k &\iff \hat{x}_{ij}^k = 0 \iff \hat{z}_{ij}^k \geq \delta \end{aligned} \tag{4.16}$$

from Proposition 3.1(a) and Assumption D(i). Let $(i', j') \in \mathcal{I}_2^k$ be chosen arbitrarily.³ Then we have

$$\varepsilon > \sum_{i=1}^2 \sum_{j=1}^m \hat{z}_{ij}^k \hat{x}_{ij}^{k+1} (\hat{e}_{ij}^k)^\top \hat{e}_{ij}^{k+1} \geq \hat{z}_{i'j'}^k \hat{x}_{i'j'}^{k+1} (\hat{e}_{i'j'}^k)^\top \hat{e}_{i'j'}^{k+1} \geq \frac{1}{2} \hat{x}_{i'j'}^{k+1} \delta,$$

where the second inequality is due to the nonnegativity of \hat{x}_{ij}^{k+1} , \hat{z}_{ij}^k and $(\hat{e}_{ij}^{k+1})^\top \hat{e}_{ij}^k$, and the last inequality follows from Proposition 3.1(c) and $(i', j') \in \mathcal{I}_2^k$. Since $\varepsilon > 0$ can be chosen arbitrarily small and we have (4.16), the above inequality means $\hat{x}_{i'j'}^{k+1} = 0$. Hence, we have $\mathcal{I}_2^k \subset \mathcal{I}_2^{k+1}$.

³ When $\mathcal{I}_2^k = \emptyset$, we immediately obtain the desired result $\mathcal{I}_2^k \subset \mathcal{I}_2^{k+1}$.

Now, by (4.15), we have

$$\begin{aligned} \gamma < \sum_{i=1}^2 \sum_{j=1}^m \hat{z}_{ij}^{k+1} \hat{x}_{ij}^k \left(\hat{e}_{ij}^{k+1}\right)^\top \hat{e}_{ij}^k &= \sum_{(i,j) \in \mathcal{I}_1^{k+1}} \hat{z}_{ij}^{k+1} \hat{x}_{ij}^k \left(\hat{e}_{ij}^{k+1}\right)^\top \hat{e}_{ij}^k \\ &+ \sum_{(i,j) \in \mathcal{I}_2^k} \hat{z}_{ij}^{k+1} \hat{x}_{ij}^k \left(\hat{e}_{ij}^{k+1}\right)^\top \hat{e}_{ij}^k \\ &+ \sum_{(i,j) \in \mathcal{I}_2^{k+1} \setminus \mathcal{I}_2^k} \hat{z}_{ij}^{k+1} \hat{x}_{ij}^k \left(\hat{e}_{ij}^{k+1}\right)^\top \hat{e}_{ij}^k \\ &= \sum_{(i,j) \in \mathcal{I}_2^{k+1} \setminus \mathcal{I}_2^k} \hat{z}_{ij}^{k+1} \hat{x}_{ij}^k \left(\hat{e}_{ij}^{k+1}\right)^\top \hat{e}_{ij}^k, \end{aligned}$$

which implies $\mathcal{I}_2^{k+1} \setminus \mathcal{I}_2^k \neq \emptyset$. Since $k \geq L$ can be chosen arbitrarily, it must hold

$$|\mathcal{I}_2^L| < |\mathcal{I}_2^{L+1}| < |\mathcal{I}_2^{L+2}| < \dots$$

However, this contradicts the boundedness of $\{|\mathcal{I}_2^k|\}$. Thus, Algorithm 1 must terminate in a finite number of iterations. □

Case 2: More General Constraint Function

When the constraint function $c(\cdot, s)$ is neither affine nor quadratic, we may consider the following perturbed problem instead of the original SOCCSIP:

$$\begin{aligned} \text{(SOCCSIP}_\varepsilon\text{)} \quad &\text{minimize } f(x) + \varepsilon \|x\|^2 \\ &\text{subject to } x \in \mathcal{K}, \quad c(x, s) \leq 0 \quad \forall s \in \Omega, \end{aligned}$$

where $\varepsilon > 0$ is a very small constant. Since the objective function of $\text{SOCCSIP}_\varepsilon$ is strictly convex, the finite convergence result in Sect. 4.2 can be applied. In this case, it is important to see that, if we choose a sufficiently small $\varepsilon > 0$, then the optimum of $\text{SOCCSIP}_\varepsilon$ is sufficiently close to the original SOCCSIP optimum. The following theorem provides the positive answer for that.

Theorem 4.3 *Let x_ε^* be the unique optimum of $\text{SOCCSIP}_\varepsilon$. Let X be the optimum set of the original SOCCSIP. Then we have $\lim_{\varepsilon \searrow 0} \text{dist}(x_\varepsilon^*, X) = 0$.*

Proof Let $\mathcal{S} : \mathbb{R} \rightarrow 2^{\mathbb{R}^n}$ be the set-valued mapping such that $\mathcal{S}(\varepsilon)$ is the solution set of $\text{SOCCSIP}_\varepsilon$. Then we have $\mathcal{S}(\varepsilon) = \{x_\varepsilon^*\}$ for all $\varepsilon > 0$ and $\mathcal{S}(0) = X$. Since it is known that the optimal set mapping \mathcal{S} is upper semi-continuous, any accumulation point of $\mathcal{S}(\varepsilon)$ is contained in $\mathcal{S}(0)$. Thus we have $\lim_{\varepsilon \searrow 0} \text{dist}(x_\varepsilon^*, X) = 0$. □

4.4 Approximation Analysis for Obtained Solution

So far, we have shown the finite termination property of Algorithm 1 via the aforementioned theorems. Nevertheless, these theorems would be meaningless if the obtained solution is far from the optimum of SOCCSIP (1.1). The following theorem guarantees that if $\eta > 0$ is sufficiently close to 0, then the last output of Algorithm 1 is also close to the optimal solution of SOCCSIP (1.1).

Theorem 4.4 *Suppose that Algorithm 1 terminates in a finite number of iterations. Let $k^*(\eta)$ be the number of iterations in which Algorithm 1 terminates. Then, $\lim_{\eta \rightarrow 0} \text{dist}(x^{k^*(\eta)}, S) = 0$.*

Proof Let $h : \mathbb{R}^n \rightarrow \mathbb{R}$, $X \subset \mathbb{R}^n$, and $\mathcal{S}_\eta \subset \mathbb{R}^n$ be defined by

$$h(x) := \max_{s \in \Omega} c(x, s), \quad X := \mathcal{K} \cap \left\{ x \mid f(x) \leq V^* \right\}, \quad \mathcal{S}_\eta := X \cap \left\{ x \mid h(x) \leq \eta \right\}.$$

Then, h is continuous and convex, X is closed and convex, and \mathcal{S}_0 coincides with the solution set of SOCCSIP (1.1). Moreover, since $f(x^{k^*(\eta)}) \leq V^*$, it follows $x^{k^*(\eta)} \in \mathcal{S}_\eta$ for any $\eta > 0$. We can show the remainder of the proof in a way analogous to [13, Thm. 3.4]. \square

Remark In Step 1 of Algorithm 1, we may also choose $l (\geq 2)$ different points $\{s_1^k, \dots, s_l^k\}$ such that $c(x^k, s_i^k) > \eta$ for $i = 1, \dots, l$ with $\bar{E}_{k+1} := E_k \cup \{s_1^k, \dots, s_l^k\}$. For such a multiple explicit exchange method, Theorems 4.2–4.4 can be shown by using analogous techniques.

5 Numerical Results

In this section, we report some numerical results. We implement Algorithm 1 by Matlab 7.10.0 (R2010a) and run the experiments on a computer with Pentium(R) CPUs 3.19GHz and 3.20GHz with 0.99MB RAM. Throughout the experiments, we set $\eta := 10^{-6}$ and $E_0 := \Omega_0$. In Step 1, we find an $s_{\text{new}}^k \in \Omega$ with $c(x^k, s_{\text{new}}^k) > \eta$ as follows. We first test $N (\approx 100)$ grid points⁴ $\bar{s}_1, \dots, \bar{s}_N \in \Omega$ to find $\bar{s}^k := \operatorname{argmax}_{i=1,2,\dots,N} c(x^k, \bar{s}_i)$. If $c(x^k, \bar{s}^k) > \eta$, then we set $s_{\text{new}}^k := \bar{s}^k$. Otherwise, we solve the constrained maximization problem: “maximize $c(x^k, s)$ subject to $s \in \Omega$ ” by means of *fmincon* solver in Matlab Optimization Toolbox with the initial point \bar{s}^k . In Step 2, we solve CSOCP(\bar{E}_{k+1}) by using the SOCCP reformulation technique together with the regularized smoothing Newton method [15]. In Step 3, we relax the criterion $v_{k+1}(s) > 0$ to $v_{k+1}(s) > 10^{-6}$. We stop the iteration of Algorithm 1 when $\max\{c(x^k, s) \mid s \in \Omega\} \leq \eta$. For all test problems, we checked that the required assumptions are satisfied.

5.1 Experiment 1 (Solving SOCCSIPs with Various Choices of Parameters)

Let $p : \mathbb{R}^2 \rightarrow \mathbb{R}$, $q : \mathbb{R}^2 \rightarrow \mathbb{R}^n$ and $r : \mathbb{R}^2 \rightarrow \mathbb{R}$ be defined as

$$p(s) := 0.1 s_1^2 (1 + \sin s_2), \quad q(s) := \left(\cos(-1)^j (s_1 s_2 + 0.5\pi) \right)_{j=1}^n + e, \\ r(s) := -(5 + \sin s_1 + \log(s_2 + 10)),$$

where $e \in \mathbb{R}^n$ denotes the identity element with respect to \mathcal{K} . (For example, $e = (1, 0, 0, 0, 1, 0, 0)^\top$ when $\mathcal{K} = \mathcal{K}^4 \times \mathcal{K}^3$). Then, we solve the following SOCCSIP:

$$\begin{aligned} &\text{minimize} \quad f(x) := \log(1 + \exp(x^\top Ax)) + b^\top x \\ &\text{subject to} \quad c(x, s) := p(s)(x^\top Mx)^{1.5} + q(s)^\top x + r(s) \leq 0 \quad \forall s \in \Omega := [-\beta\pi, \beta\pi]^2 \\ &\quad \quad \quad x \in \mathcal{K}, \end{aligned} \tag{5.1}$$

where $\beta > 0$ is a given constant, and $A \in \mathbb{R}^{n \times n}$ and $M \in \mathbb{R}^{n \times n}$ are positive semidefinite symmetric matrices. Function f is convex, but is not strictly convex when $\operatorname{rank}(A) < n$. Also c is convex with respect to x , but is not strictly convex when $\operatorname{rank}(M) < n$. Therefore, we replace the objective function by $f(x) + 10^{-10} \|x\|^2$ so that the finite convergence is

⁴ When $\Omega = [l_s, u_s] \subset \mathbb{R}$, we test the 101 points $l_s + i(u_s - l_s)/100$ with $i = 0, 1, \dots, 100$. When $\Omega = [l_s, u_s]^2 \subset \mathbb{R}^2$, we test the 121 points $(l_s + i(u_s - l_s)/10, l_s + j(u_s - l_s)/10)^\top$ with $i = 0, 1, \dots, 10$ and $j = 0, 1, \dots, 10$.

Table 1 Obtained results for SOCCSIP (5.1) with various choices of β

Problem	β	Output ($\mathcal{K} = \mathcal{K}^{10}, r_m = 8, r_a = 6$)				
		λ_1	λ_2	#ite	cpu(s)	$E_k^{\text{final}} \setminus \Omega_0$
Problem 1	0.1	0	5.97	1	0.39	$(-0.1\pi, 0.1\pi)^\top$
Problem 2	0.1	0	8.29	1	0.50	$(-0.1\pi, 0.1\pi)^\top$
Problem 3	0.1	0	8.13	1	0.85	$(-0.1\pi, 0.1\pi)^\top$
Problem 4	1	0	0.95	0	0.21	\emptyset
Problem 5	1	0	1.41	5	2.96	$(-\pi, 0.492\pi)^\top, (\pi, 0.492\pi)^\top$
Problem 6	1	0	1.63	2	0.54	$(\pi, 0.483\pi)^\top$
Problem 7	1.6	0	1.37	4	4.80	$(1.592\pi, -1.305\pi)^\top$
Problem 8	1.6	0	1.39	6	6.85	$(1.6\pi, -1.550\pi)^\top$
Problem 9	1.6	0	0.54	0	0.11	\emptyset
Problem 10	2	0	1.42	8	6.91	$(-2\pi, -1.567\pi)^\top$
Problem 11	2	0	1.08	5	6.20	$(2\pi, -1.636\pi)^\top$
Problem 12	2	0	2.06	4	1.22	$(2\pi, -1.652\pi)^\top$

obtained theoretically. Matrices A and M are defined as $A := PP^\top$ and $M := QQ^\top$, where P and Q are respectively $(n \times r_a)$ - and $(n \times r_m)$ -dimensional matrices whose components are randomly chosen from $[-1, 1]$. Since P and Q are randomly generated, we almost always have $\text{rank}(A) = \min(n, r_a)$ and $\text{rank}(M) = \min(n, r_m)$. Also, each component of vector b is randomly chosen from $[-1, 1]$. In applying Algorithm 1, we set $\Omega_0 := \{0\}$ since $\mathcal{K} \cap \{x \mid c(x, 0) \leq 0\}$ is compact. SOCCSIP (5.1) has a Slater point since $c(0, s) = r(s) < 0$ for all $s \in \Omega$.

First, we solve SOCCSIP (5.1) with various choices of constant $\beta > 0$. We solve 12 problem instances, each of which has different values of A, M and b . For all instances, we set $\mathcal{K} = \mathcal{K}^{10}, r_a = 6$ and $r_m = 8$. Since A and M are rank-deficient, functions f and $c(\cdot, s)$ are not strictly convex. We show the obtained results in Table 1, in which λ_1 and λ_2 denote the spectral values defined by (2.1) of the obtained solutions x^* , #ite denotes the number of iterations, cpu(s) denotes the CPU time in seconds, and $E_k^{\text{final}} \setminus \Omega_0$ denotes the final output of E_k except Ω_0 . Notice that the number of #ite does not count the first subproblem CSOCP(E_0). Therefore, Algorithm 1 actually solves #ite + 1 CSOCPs for each instance. The rows with #ite = 0 (Problems 4 and 9) imply that Algorithm 1 finds the SOCCSIP optimum $x^* (= x^0)$ in Step 0, and there exists no $s_{\text{new}}^0 \in \Omega$ with $c(x^0, s_{\text{new}}^0) > \eta$. In other words, the convex constraint corresponding to $s = (0, 0)^\top$ is active at the solution x^* . We can also see that, when $\beta = 0.1$ (Problems 1–3), the algorithm finds the solution x^* with $k = 1$, and the final active index is $s = (-0.1\pi, 0.1\pi)^\top$. Notice that Ω is expressed as a two-dimensional square, and $(-0.1\pi, 0.1\pi)^\top$ is one of its vertices. Since Algorithm 1 checks the values of $c(x^k, s)$ at all the vertices in Step 1, we will find the SOCCSIP optimum very soon if the final active index is located at the vertex of Ω . On the contrary, for Problems 5–8 and 10–12, the final active indices are not located at the vertices of Ω .⁵ This may be the main reason

⁵ For Problem 7, the final active index is located in the interior of Ω , and for other six problems (5, 6, 8, 10, 11, 12), they are located on the non-vertex boundary.

Table 2 Obtained results for SOCCSIP (5.1) with various choices of \mathcal{K}

n	\mathcal{K}	Output ($r_m = r_a = 0.8n, \beta = 1$)							
		λ_1^{\min}	λ_1^{\max}	$\lambda_1^{\text{zero}} (\%)$	λ_2^{\min}	λ_2^{\max}	$\lambda_2^{\text{zero}} (\%)$	#ite	cpu(s)
10	$(\mathcal{K}^1)^{10}$	0	0.98	51.2	–	–	–	1.70	1.41
10	$(\mathcal{K}^2)^5$	0	0.91	79.2	0	1.46	27.0	1.37	0.98
10	\mathcal{K}^{10}	0	0	100	0.12	2.36	0	1.44	0.84
100	$(\mathcal{K}^{20})^5$	0	0	100	0.06	0.32	0	2.12	2.64
100	$(\mathcal{K}^{50})^2$	0	0	100	0.17	0.45	0	2.12	2.61
100	\mathcal{K}^{100}	0	0	100	0.27	0.63	0	2.24	2.98
200	$(\mathcal{K}^{20})^{10}$	0	0	100	0.02	0.17	0	2.37	11.63
200	$(\mathcal{K}^{50})^4$	0	0	100	0.07	0.24	0	2.09	10.76
200	\mathcal{K}^{200}	0	0	100	0.27	0.38	0	2.05	13.97

why Problems 5–8 and 10–12 need more iterations and cpu time than Problems 1–3. This tendency seems to be more noticeable as β becomes larger.

Next, we solve SOCCSIP (5.1) with various choices of the Cartesian structure \mathcal{K} . We consider 9 different Cartesian structures, and solve 100 problems for each \mathcal{K} . We therefore solve 900 problems in total. For all problems, we set $r_a = r_m = 0.8n$ and $\beta = 1$. Since A and M are rank-deficient, functions f and $c(\cdot, s)$ are not strictly convex. We give the obtained results in Table 2, in which λ_i^{\min} and λ_i^{\max} ($i = 1, 2$) denote the maximum and minimum of the spectral values (2.1) among all Cartesian subvectors x_1^*, \dots, x_m^* in 100 problems for each \mathcal{K} .⁶ (In case of $\mathcal{K} = (\mathcal{K}^1)^{10} = \mathbb{R}_+^{10}$, we only give the values for λ_1). Also, λ_i^{zero} denotes the frequency that the spectral values become 0, and #ite and cpu(s) are the average values among the 100 trials for each \mathcal{K} . From the table, we can observe that the number of iterations does not change so much even when the dimension n of the variables or the number m of sub-SOCs increases. However, CPU time increases quite a bit as n becomes larger. This implies that the computational cost for solving each subproblem (CSOCs solved in Steps 0 and 2) becomes more expensive as n increases. Also, we can see that the spectral value λ_1 often becomes 0, which implies that each subvector x_j^* is located on the boundary of the SOC \mathcal{K}^{nj} . When $\mathcal{K} = (\mathcal{K}^1)^{10} = \mathbb{R}_+^{10}$, approximately 50 % of obtained λ_1 s are greater than 0, but it is not surprising since, in this case, each subvector x_j^* coincides with the j -th scalar component of vector x^* , and $\lambda_1(x_j^*) > 0$ implies $x_j^* > 0$. When $\mathcal{K} = (\mathcal{K}^2)^5$, we still have more than 20 % positive λ_1 s. It is also convincing since the dimension of each SOC is small ($= 2$) and the problems are generated randomly. In other seven cases with $\mathcal{K} = \mathcal{K}^{10}, (\mathcal{K}^{20})^5, \dots, \mathcal{K}^{200}$, we always have $\lambda_1(x_j^*) = 0$ and $\lambda_2(x_j^*) > 0$, which means that all the subvectors of x^* are located on the boundary of SOCs.

Finally, we solve SOCCSIP (5.1) with various degrees of rank deficiency of matrices A and M . For all problems, we set $\mathcal{K} = \mathcal{K}^{20} \times \mathcal{K}^{30}, \beta = 1$, and $r := r_a = r_m$, and choose 7 different values for r . We solve 100 problems for each r , and hence solve 700 problems in

⁶ For example, when $\mathcal{K} = (\mathcal{K}^\ell)^m$, we have m subvectors $x_1^*, \dots, x_m^* \in \mathbb{R}^\ell$ for the optimum x^* . Therefore, if the obtained solutions of the 100 problems are $x^{*,1}, x^{*,2}, \dots, x^{*,100}$, then we have $\lambda_i^{\max} := \max \{ \lambda_i(x_j^{*,p}) \mid (j, p) \in \{1, 2, \dots, m\} \times \{1, 2, \dots, 100\} \}$ and $\lambda_i^{\min} := \min \{ \lambda_i(x_j^{*,p}) \mid (j, p) \in \{1, 2, \dots, m\} \times \{1, 2, \dots, 100\} \}$ for each $i = 1, 2$.

Table 3 Obtained results for SOCCSIP (5.1) with various choices of $r := r_a = r_m$

r	Output ($\mathcal{K} = \mathcal{K}^{20} \times \mathcal{K}^{30}, \beta = 1$)							
	λ_1^{\min}	λ_1^{\max}	$\lambda_1^{\text{zero}} (\%)$	λ_2^{\min}	λ_2^{\max}	$\lambda_2^{\text{zero}} (\%)$	#ite	cpu(s)
1	0	0	100	0	12.03	30.0	12.56	12.11
2	0	0	100	0	12.03	19.0	13.01	11.65
5	0	0	100	0	11.99	2.0	13.20	10.31
10	0	0	100	0.859	8.91	0	8.80	7.08
20	0	0	100	0.181	4.50	0	2.88	2.68
50	0	0	100	0.085	0.56	0	1.16	0.96
100	0	0	100	0.031	0.27	0	0.03	0.27

total. Note that we usually have $\text{rank}(A) = \text{rank}(M) = \min(50, r)$. Therefore, f and $c(\cdot, s)$ are not strictly convex when $r < 50$, and are (almost always) strictly convex when $r \geq 50$. We give the obtained results in Table 3, in which $\lambda_i^{\min}, \lambda_i^{\max}$ and λ_i^{zero} are defined analogously to the previous experiment, and #ite and cpu(s) are the average of 100 problems for each r . As the table shows, we need a large number of iterations when r is small, i.e., matrices A and M have high rank-deficiency. On the other hand, when $r = 50$ and 100, i.e., f and $c(\cdot, s)$ are strictly convex, we obtain the solution in a very small number of iterations. Especially, when $r = 100$, we often obtain the SOCCSIP optimum in the initial step with $k = 0$. Also we can observe that, when r is small, the value of λ_2 sometimes becomes 0, which implies that the optimal solution of SOCCSIP (5.1) is $x^* = 0$. Actually, the above-mentioned features generally depend on the structure of each problem, but in many cases the ill-posedness of a problem seems to be relevant to the rank deficiency of certain matrices involved in the problem.

5.2 Experiment 2 (Application to Robust Optimization)

The robust optimization [2] is one of distribution-free methodologies for handling problems with uncertain data. We usually assume that the uncertain data belong to some set, and try to solve another optimization problem called robust counterpart (RC), which is composed with taking the worst possible case into consideration.

Algorithm 1 is also applicable to the robust optimization for convex semi-infinite programs (CSIPs). Consider the following uncertain CSIP:

$$\begin{aligned} & \text{minimize } \hat{b}^\top x \\ & \text{subject to } c(x, s) \leq 0 \quad \forall s \in \Omega, \end{aligned} \tag{5.2}$$

where $\Omega := [-1, 1]$ and $c(x, s) := x^\top \Xi(s)x + \eta(s)^\top x + \zeta(s)$ with

$$\begin{aligned} \Xi(s) &:= \begin{bmatrix} 19 & 3 & -6 & -7 & 5 \\ 3 & 18 & -5 & 2 & -4 \\ -6 & -5 & 15 & 4 & -5 \\ -7 & 2 & 4 & 10 & -2 \\ 5 & -4 & -5 & -2 & 16 \end{bmatrix} + \text{diag}(\sin(\alpha s))_{\alpha=1}^5 \\ \eta(s) &:= (4, -3, 1, -2, 4)^\top + (\cos 1.3\alpha s)_{\alpha=1}^5, \quad \zeta(s) := -10 + (5 + s)^{-1}. \end{aligned}$$

Table 4 Obtained results for SOCCSIP (5.4) with various choices of ρ

ρ	Output ($b = (4, 9, -6, 8, -5)^\top$, $\mathcal{K} = \mathcal{K}^6$)					λ_1	λ_2	#ite	cpu(s)
	$u^{*,\rho}$	$x^{*,\rho}$							
0	1.167	$(-0.567, 0.076, 0.394, -0.875, 0.336)^\top$	0	2.334	2	1.768			
0.5	1.147	$(-0.555, 0.054, 0.391, -0.861, 0.331)^\top$	0	2.295	2	2.089			
1	1.126	$(-0.542, 0.031, 0.388, -0.845, 0.326)^\top$	0	2.252	3	3.485			
1.5	1.103	$(-0.529, 0.008, 0.384, -0.828, 0.321)^\top$	0	2.207	4	1.748			
2	1.079	$(-0.514, -0.015, 0.380, -0.810, 0.315)^\top$	0	2.159	3	1.625			
2.5	1.054	$(-0.499, -0.040, 0.376, -0.790, 0.310)^\top$	0	2.109	2	1.268			
3	1.028	$(-0.483, -0.065, 0.371, -0.768, 0.303)^\top$	0	2.057	2	1.014			

Moreover, $\hat{b} \in \mathbb{R}^n$ is an uncertain vector such that $\hat{b} = b + \delta_b$, where $b = (-3, 4, 2, -4, 1)^\top$ is the nominal value of \hat{b} , and δ_b is the error term belonging to a certain set \mathcal{B} . Then, the RC of CSIP (5.2) is written as

$$\begin{aligned} & \text{minimize } \max_{\delta_b \in \mathcal{B}} (b + \delta_b)^\top x \\ & \text{subject to } c(x, s) \leq 0 \quad \forall s \in \Omega. \end{aligned} \tag{5.3}$$

Now, suppose that \mathcal{B} is a closed sphere with radius ρ , that is, $\mathcal{B} := \{\delta_b \mid \|\delta_b\| \leq \rho\}$. Then, the objective function of RC (5.3) can be calculated as $\max_{\delta_b \in \mathcal{B}} (b + \delta_b)^\top x = b^\top x + \max\{\delta_b^\top x \mid \|\delta_b\| \leq \rho\} = b^\top x + \rho \|x\|$. Therefore, by introducing an auxiliary variable $u \in \mathbb{R}$, RC (5.3) can be rewritten equivalently as

$$\begin{aligned} & \text{minimize}_{x,u} b^\top x + \rho u \\ & \text{subject to } \|x\| \leq u, \quad c(x, s) \leq 0 \quad \forall s \in \Omega, \end{aligned} \tag{5.4}$$

which is of the form SOCCSIP (1.1) with $x := \begin{pmatrix} x \\ u \end{pmatrix}$ and $\mathcal{K} := \mathcal{K}^{n+1}$.

Table 4 shows the obtained results on SOCCSIP (5.4) with various choices of ρ . In the table, $u^{*,\rho}$ and $x^{*,\rho}$ are the solutions of SOCCSIP (5.4), λ_1 and λ_2 are the spectral values for $\begin{pmatrix} u^{*,\rho} \\ x^{*,\rho} \end{pmatrix}$, and #ite and cpu(s) denote the number of iterations and CPU time in seconds for Algorithm 1, respectively. As the table shows, the solution of RC (5.3) moves continuously as the radius ρ of \mathcal{B} varies. Also, for all cases, we have $\lambda_1 = 0$, i.e., $u^{*,\rho} = \|x^{*,\rho}\|$, and the algorithm finds the solution in a small number of iterations and CPU time.

Next, we investigate the distribution of the functional values to observe the actual effect of the robust optimization. We generate 10,000 sample vectors $\delta_b^1, \delta_b^2, \dots, \delta_b^{10000} \in \mathbb{R}^5$ for the error term δ_b . Each δ_b^i is defined as $\delta_b^i := \gamma_i v^i / \|v^i\|$, where $\gamma_i \in \mathbb{R}$ and each component of $v^i \in \mathbb{R}^5$ independently follow the normal distribution with mean 0 and deviation 1.5. Since the normal distribution contains 95 % of the values within 2 standard deviations of the mean, we will have $\|\delta_b^i\| \leq 3$ with 95 % probability. To make the functional values more intuitive, we add a positive constant (= 18) to the objective function, that is, we check the value of $f_i(x^{*,\rho}) := (b + \delta_b^i)^\top x^{*,\rho} + 18$ for each i and ρ . The obtained results are summarized in Table 5, where the columns of ‘best’, ‘mean’ and ‘worst’ denote the minimum, average, and maximum of $f_i(x^{*,\rho})$ among $i = 1, 2, \dots, 10,000$ for each ρ , respectively. The column of [6.5, 7) denotes the number of times that we had $6.5 \leq f_i(x^{*,\rho}) < 7$ among 10,000 sample vectors of δ_b^i . (Other columns are similar). From the table, we can see that the values of ‘worst’

Table 5 Perturbed functional values of uncertain CSIP (5.2)

ρ	Output $(f_i(x^{*,\rho}) = (b + \delta_b^i)^\top x^{*,\rho} + 18)$								
	Best	Mean	Worst	[6.5, 7)	[7, 7.5)	[7.5, 8)	[8, 8.5)	[8.5, 9)	[9, +∞)
0	0.464	5.358	9.967	342	162	87	45	12	4
0.5	0.581	5.363	9.864	331	162	86	39	10	4
1	0.717	5.379	9.763	337	160	83	37	9	4
1.5	0.770	5.408	9.666	348	155	87	34	8	4
2	0.813	5.450	9.574	357	163	86	36	6	4
2.5	0.877	5.506	9.488	382	173	89	34	7	3
3	0.961	5.578	9.410	438	182	91	38	7	2

is smaller as ρ becomes larger, though it is opposite in the columns of ‘best’ and ‘mean’. This means that the robust optimization may have disadvantage under average or lucky situations, but the serious damage can be avoided or reduced even when an unlucky situation occurs. Since we applied the normal distribution to generate the sample error vectors, we seldom encountered the unlucky situations such as $f_i(x^{*,\rho}) > 8.5$. However, if we apply another type of distribution, we may encounter such an undesirable situation more often, and the robust optimization can be more important.

References

1. Auslender, A., Goberna, M.A., López, M.A.: Penalty and smoothing methods for convex semi-infinite programming. *Math. Oper. Res.* **34**, 303–319 (2009)
2. Ben-Tal, A., El Ghaoui, L., Nemirovski, A.: *Robust Optimization*. Princeton University Press, Princeton (2009)
3. Ben-Tal, A., Nemirovski, A.: *Robust convex optimization*. *Math. Oper. Res.* **23**, 769–805 (1998)
4. Ben-Tal, A., Nemirovski, A.: Robust solutions of uncertain linear programs. *Oper. Res. Lett.* **25**, 1–13 (1999)
5. Bertsekas, D.P., Nedić, A., Ozdaglar, A.E.: *Convex Analysis and Optimization*. Athena Scientific, Belmont (2003)
6. Bonnans, J.F., Ramírez, H.: Perturbation analysis of second-order cone programming problems. *Math. Progr.* **104**, 205–227 (2005)
7. Bonnans, J.F., Shapiro, A.: *Perturbation analysis of optimization problems*. Springer, New York (2000)
8. Chen, X.D., Sun, D., Sun, J.: Complementarity functions and numerical experiments for second-order cone complementarity problems. *Comput. Optim. Appl.* **25**, 39–56 (2003)
9. Faraut, J., Korányi, A.: *Analysis on Symmetric Cones*. Clarendon Press, New York (1994)
10. Fukushima, M., Luo, Z.-Q., Tseng, P.: Smoothing functions for second-order cone complementarity problems. *SIAM J. Optim.* **12**, 436–460 (2001)
11. Goberna, M.A., López, M.A.: Linear semi-infinite programming: an updated survey. *Eur. J. Oper. Res.* **143**, 390–405 (2002)
12. Gonzaga, C., Polak, E., Trahan, R.: An improved algorithm for optimization problems with functional inequality constraints. *IEEE Trans. Autom. Control* **25**, 49–54 (1980)
13. Hayashi, S., Wu, S.-Y.: An explicit algorithm for linear semi-infinite programming problem with second-order cone constraints. *SIAM J. Optim.* **20**, 1527–1546 (2009)
14. Hayashi, S., Yamaguchi, T., Yamashita, N., Fukushima, M.: A matrix splitting method for symmetric affine second-order cone complementarity problems. *J. Comput. Appl. Math.* **175**, 335–353 (2005)
15. Hayashi, S., Yamashita, N., Fukushima, M.: A combined smoothing and regularization method for monotone second-order cone complementarity problems. *SIAM J. Optim.* **15**, 593–615 (2005)
16. Hettich, R., Kortanek, K.O.: Semi-infinite programming: theory, methods, and applications. *SIAM Rev.* **35**, 380–429 (1993)

17. Kato, H., Fukushima, M.: An SQP-type algorithm for nonlinear second-order cone programs. *Optim. Lett.* **1**, 129–144 (2007)
18. Kortanek, K., No, H.: A central cutting plane algorithm for convex semi-infinite programming problems. *SIAM J. Optim.* **3**, 901–918 (1993)
19. Lai, H.C., Wu, S.-Y.: On linear semi-infinite programming problems: an algorithm. *Numer. Funct. Anal. Optim.* **13**, 287–304 (1992)
20. Li, D., Qi, L., Tam, J., Wu, S.-Y.: A smoothing newton method for semi-infinite programming. *J. Glob. Optim.* **30**, 169–194 (2004)
21. Lin, C.-J., Fang, S.-C., Wu, S.-Y.: An unconstrained convex programming approach to linear semi-infinite programming. *SIAM J. Optim.* **8**, 443–456 (1998)
22. Lobo, M.S., Vandenberghe, L., Boyd, S., Lebret, H.: Applications of second-order cone programming. *Linear Algebra Appl.* **284**, 193–228 (1998)
23. López, M., Still, G.: Semi-infinite programming. *Eur. J. Oper. Res.* **180**, 491–518 (2007)
24. Narushima, Y., Sagara, N., Ogasawara, H.: A smoothing Newton method with Fischer-Burmeister function for second-order cone complementarity problems. *J. Optim. Theory Appl.* **149**, 79–101 (2011)
25. Okuno, T., Hayashi, S., Fukushima, M.: A regularized explicit exchange method for semi-infinite programs with an infinite number of conic constraints. *SIAM J. Optim.* **22**, 1009–1028 (2012)
26. Özmen, A., Weber, G.-W., Batmazb, I., Kropat, E.: RCMARS: robustification of CMARS with different scenarios under polyhedral uncertainty set. *Commun. Nonlinear Sci. Numer. Simul.* **16**, 4780–4787 (2011)
27. Qi, L., Wu, S.-Y., Zhou, G.: Semismooth newton for solving semi-infinite programming problems. *J. Glob. Optim.* **27**, 215–232 (2003)
28. Reemtsen, R., Görner, S.: Numerical methods for semi-infinite programming: a survey. In: Reemtsen, R., Rückmann, J. (eds.) *Semi-Infinite Programming*, pp. 195–275. Kluwer Academic Publishers, Boston (1998)
29. Reemtsen, R., Rückmann, J.: *Semi-Infinite Programming*. Kluwer Academic Publishers, Boston (1998)
30. Stein, O., Still, G.: Solving semi-infinite optimization problems with interior point techniques. *SIAM J. Control Optim.* **42**, 769–788 (2003)
31. Still, G.: Discretization in semi-infinite programming: the rate of convergence. *Math. Progr.* **91**, 53–69 (2001)
32. Wu, S.-Y., Fang, S.-C.: Solving convex programs with infinitely linear constraints by a relaxed cutting plane method. *Comput. Math. Appl.* **38**, 23–33 (1999)
33. Wu, S.-Y., Li, D.H., Qi, L., Zhou, G.: An iterative method for solving KKT system of the semi-infinite programming. *Optim. Methods Softw.* **20**, 629–643 (2005)
34. Yamashita, H., Yabe, H.: A primal-dual interior point method for nonlinear optimization over second-order cones. *Optim. Methods Softw.* **24**, 407–426 (2009)
35. Yuan, Y.X.: Recent advances in numerical methods for nonlinear equations and nonlinear least squares. *Numer. Algebra Control Optim.* **1**, 15–34 (2011)
36. Yuan, Y.X.: A trust region algorithm for Nash equilibrium problems. *Pac. J. Optim.* **7**, 125–138 (2011)
37. Zhang, L., Wu, S.-Y., López, M.A.: A new exchange method for convex semi-infinite programming. *SIAM J. Optim.* **20**, 2959–2977 (2010)