An explicit exchange algorithm for linear semi-infinite programming problems with second-order cone constraints[*]

Shunsuke Hayashi[†] and Soon-Yi Wu[‡]

Abstract   In this paper, we propose an explicit exchange algorithm for solving semi-infinite programming problem (SIP) with second-order cone (SOC) constraints. We prove, by using the slackness complementarity conditions, that the algorithm terminates in a finite number of iterations and the obtained solution sufficiently approximates the original SIP solution. In existing studies on SIPs, only the nonnegative constraints were considered, and hence, the slackness complementarity conditions were separable to each component. However, in our study, the existing componentwise analyses are not applicable anymore since the slackness complementarity conditions are associated with SOCs. In order to overcome such a difficulty, we introduce a certain coordinate system based on the spectral factorization. In the numerical experiments, we solve some test problems to see the effectiveness of the proposed algorithm.

1   Introduction

A semi-infinite programming problem (SIP) [2, 12, 16, 17, 18, 29] is to minimize an objective function with a finite-dimensional argument under an infinite number of constraints. Generally, SIP is written as follows:

$$
\begin{aligned}
&\text{Minimize} \quad f(\boldsymbol{x}) \\
&\text{subject to} \quad \boldsymbol{x} \in X, \quad g(\boldsymbol{x}, t) \geq 0 \ \ (\forall t \in T),
\end{aligned}
\tag{1.1}
$$

where $f : \Re^n \to \Re$ and $g : \Re^n \times T \to \Re$ are given functions, $X \subseteq \Re^n$ is a given convex set, and T is a given compact Hausdorff space. If T is a finite set, the problem (1.1) reduces to a finite optimization problem such as nonlinear programming problem (NLP) or linear programming problem (LP) . The SIP plays an important role in various fields such as approximation theory, optimal control, numerous engineering problems, etc [11, 21, 23, 24].

Many approaches have been studied for solving SIPs. One of important methods is the discretization method [21, 24, 26], in which the infinite set T is approximated by a sequence of finite subsets $\{T^k\}$ such that the distance[1] from T^k to T tends to 0 as the iteration k goes infinity. For a sufficiently

---

[†]Department of Applied Mathematics and Physics, Graduate School of Informatics, Kyoto University, Kyoto 606-8501, Japan (shunhaya@amp.i.kyoto-u.ac.jp, +81-75-753-4758)

[‡]Department of Mathematics, National Cheng Kung University / National Center for Theoretical Sciences, Tainan, Taiwan, R.O.C. (soonyi@mail.ncku.edu.tw, +886-6-2757575 Ext. 65080)

[1]For two sets Y and $Y_\sigma \subseteq Y$, their distance is defined by $\mathrm{dist}\,(Y_\sigma, Y) := \sup_{y \in Y} \inf_{y_\sigma \in Y_\sigma} \|y - y_\sigma\|$.

large k, however, a general discretization method may become very expensive, since the number of elements in T^k can be huge, and it takes very long time to verify the feasibility with respect to the approximated subproblem. The local reduction based method was introduced by [13, 16]. It solves, in each iteration, a locally reduced problem with finite constraints, which is defined by using certain implicit functions associate with local minimizers of $g(\boldsymbol{x}, \cdot)$ over T. Discretization SQP type methods and trust region type methods based on the local reduction have been studied by some authors [6, 19, 25]. Exchange method [16, 28] is also well-known. It solves, in each iteration, the relaxed version of SIP (1.1) with T replaced by a finite set T^k. Then, $\{T^k\}$ is updated so that $T^{k+1} \subseteq T^k \cup \{t_1, \ldots, t_r\}$ where $\{t_1, \ldots, t_r\} \subseteq T \setminus T^k$. Similar to exchange method, [29] proposed an iterative method for solving KKT system of the SIP in which they drop some redundant points at some iterations. For the linear SIP, Lai and Wu [18] proposed the explicit algorithm in which they drop out redundant points in T^k for each k and only keep active points. This will ensure $|T^k| \leq n$ for each k, and hence the algorithm is very efficient in saving computational time. Lai and Wu's explicit algorithm is also a kind of exchange method.

In this paper, we focus on the linear SIP with second-order cone constraints (LSSIP) of the form

$$
\begin{aligned}
\text{Minimize} \quad & \boldsymbol{c}^\top \boldsymbol{x} \\
\text{subject to} \quad & \boldsymbol{x} \in \mathcal{K}^n, \quad \boldsymbol{a}(t)^\top \boldsymbol{x} - b(t) \geq 0 \ \ (\forall t \in T),
\end{aligned}
\tag{1.2}
$$

where vector $\boldsymbol{c} \in \Re^n$, functions $b : T \to \Re$ and $\boldsymbol{a} : T \to \Re^n$ are given, and $\mathcal{K}^n \subset \Re^n$ is the n-dimensional second-order cone (SOC) defined by

$$
\mathcal{K}^n := \left\{ \boldsymbol{x} = (x_1, \tilde{\boldsymbol{x}}) \in \Re \times \Re^{n-1} \ \middle| \ x_1 \geq \|\tilde{\boldsymbol{x}}\| \right\}.
$$

Throughout the paper, $\|\cdot\|$ denotes the Euclidean norm, and a vector $\boldsymbol{x} \in \Re^n$ is described as $\boldsymbol{x} := (x_1, \tilde{\boldsymbol{x}}) \in \Re \times \Re^{n-1}$ with $\tilde{\boldsymbol{x}} := (x_2, \ldots, x_n)^\top \in \Re^{n-1}$. For $n = 1$, $\mathcal{K}^1$ is the set of nonnegative reals, i.e., $\mathcal{K}^1 = \Re_+$.

In this decade, the problems with SOC constraints attract much attention. For instance, the linear second-order cone programming problem (SOCP), which involves some classes of convex quadratic programming problems as special cases, is solved by the interior point method efficiently [1, 20]. Especially, the primal-dual interior point method for the linear programming problems (LP) was extended to the linear SOCP by some researchers. Also, some softwares have been proposed [27]. However, the simplex type method is not popular for linear SOCP since, on the contrary to LP, the feasible region is not polyhedral in general. Although some simplex type approach for linear SOCP was studied in [22], it is applicable only to a restricted class of SOCP, and generally requires an infinite number of iterations to converge. On the other hand, nonlinear SOCP is more complicated, and there exist fewer studies. In [30], the authors proposed an algorithm which involves a primal-dual merit function that consists of the barrier function and the potential function, and showed its global convergence. The second-order cone complementarity problem (SOCCP) is an extension of the linear and nonlinear complementarity problem [7, 8] to SOC version. In [10], the authors introduced Jordan algebra to SOCCP, and analyzed some properties for the smoothing functions that are necessary for developing efficient algorithms. Based on the analyses in [10], Hayashi, Yamashita and Fukushima [15] proposed a globally and quadratically convergent algorithm with smoothing and regularization methods. Chen, Chen and Tseng [5] studied some properties of nonsmooth vector-valued functions for SOCCP, which are helpful for analyzing the convergence properties of algorithms. For linear SOCCP,

a matrix splitting method was proposed in [14], which is efficient for SOCCPs with large-scaled and sparse matrices.

The main purpose of the paper is to construct an algorithm for solving LSSIP (1.2) and study its convergence properties. Actually, we extend Lai and Wu's explicit algorithm [18] for linear SIP to LSSIP (1.2). However, its analysis technique is quite different. While the slackness complementarity conditions for each subproblem was analyzed componentwisely in [18], those for LSSIP (1.2) cannot be decomposed to each component. We therefore define the coordinate system based on the spectral factorization associated with SOCs. We also note that our algorithm introduces the relaxed scheme which does not require solving the global minimization problem with respect to $t \in T$ in each iteration. Instead of solving such a minimization problem, the algorithm has only to find some $t \in T$ such that a certain criterion with small $\gamma > 0$ is satisfied. We prove that the algorithm outputs an approximate solution in a finite number of iterations, and the obtained approximate solution converges to the original LSSIP solution as γ tends to 0.

One may think the LSSIP of the form (1.2) is too restrictive. We can indeed define a more general LSSIP in which a finite number of equality constraints are added, and the SOC constraint is replaced by the Cartesian version: $\boldsymbol{x} \in \mathcal{K}^{n_1} \times \cdots \times \mathcal{K}^{n_m}$ with $n_1 + \cdots + n_m = n$. Such an LSSIP involves every linear SOCP as a subproblem, and several applications such as robust optimization [3, 4] can be considered. However, in the paper, we focus only on a simple LSSIP of the form (1.2) to avoid complicating the analyses too much. We believe that the obtained results and proof techniques will be a foothold for the studies of more general LSSIPs.

This paper is organized as follows. In Section 2, we give some preliminaries. We first introduce the spectral factorization associated with SOCs, which is a fundamental and important tool for analyzing the SOCs. Next we give the duality theorem and the slackness complementarity conditions for LSSIP (1.2). Since LSSIP (1.2) has an infinite number of constraints, the decision variable for the dual problem cannot be written as a finite-dimensional vector. In Section 3, we propose an explicit exchange algorithm for LSSIP (1.2), and show the finite termination and some convergence result. In Section 4, we give some numerical results to see the performance of the algorithm. In Section 5, we conclude the paper with some remarks.

2 Preliminaries

In this section, we give some preliminaries that are necessary for analyzing and understanding the proposed algorithms.

2.1 Spectral factorization associated with second-order cones

We first introduce the spectral factorization for SOCs. This technique is one topic of Jordan algebra [9], and plays an important role in describing the relation between the SOC and the current iteration point in the algorithm.

For an arbitrary vector $\boldsymbol{x} := (x_1, \tilde{\boldsymbol{x}}) \in \Re \times \Re^{n-1}$ with $n \geq 2$, its spectral values $\lambda_1(\boldsymbol{x}), \lambda_2(\boldsymbol{x}) \in \Re$

and spectral vectors $\boldsymbol{v}_1(\boldsymbol{x})$, $\boldsymbol{v}_2(\boldsymbol{x}) \in \Re^n$ are defined by

$$\lambda_i(\boldsymbol{x}) := x_1 + (-1)^i \|\tilde{\boldsymbol{x}}\|,$$

$$\boldsymbol{v}_i(\boldsymbol{x}) := \begin{cases} \dfrac{1}{2}\left(1, (-1)^i \dfrac{\tilde{\boldsymbol{x}}}{\|\tilde{\boldsymbol{x}}\|}\right) & \text{if } \tilde{\boldsymbol{x}} \neq \boldsymbol{0}, \\ \dfrac{1}{2}\left(1, (-1)^i \boldsymbol{w}\right) & \text{if } \tilde{\boldsymbol{x}} = \boldsymbol{0}, \end{cases} \qquad (i = 1, 2)$$

respectively, where $\boldsymbol{w} \in \Re^{n-1}$ is an arbitrary vector such that $\|\boldsymbol{w}\| = 1$. Then the spectral factorization for $\boldsymbol{x}$ is given by

$$\boldsymbol{x} = \lambda_1(\boldsymbol{x})\boldsymbol{v}_1(\boldsymbol{x}) + \lambda_2(\boldsymbol{x})\boldsymbol{v}_2(\boldsymbol{x}). \tag{2.1}$$

Note that $\lambda_1(\boldsymbol{x}) \leq \lambda_2(\boldsymbol{x})$ always holds, and $\lambda_1(\boldsymbol{x}) \geq 0$ if and only if $\boldsymbol{x} \in \mathcal{K}^n$. Moreover, $\lambda_1(\boldsymbol{x}) = 0$ if and only if $\boldsymbol{x} \in \operatorname{bd}\mathcal{K}^n$ (boundary of $\mathcal{K}^n$), and $\lambda_1(\boldsymbol{x}) > 0$ if and only if $\boldsymbol{x} \in \operatorname{int}\mathcal{K}^n$ (interior of $\mathcal{K}^n$). We also have $\|\boldsymbol{v}_1(\boldsymbol{x})\| = \|\boldsymbol{v}_2(\boldsymbol{x})\| = 1/\sqrt{2}$ and $\boldsymbol{v}_1(\boldsymbol{x})^\top \boldsymbol{v}_2(\boldsymbol{x}) = 0$.

2.2 Duality theory and complementarity slackness conditions

Next, we introduce the duality theory for LSSIP (1.2), and the complementarity slackness conditions. Let $M(T)$ be the space of all bounded regular Borel measures on T, and $M_+(T)$ be the nonnegative cone in $M(T)$. Then, the dual problem for LSSIP (1.2) is given by

$$\begin{aligned} \text{Maximize} \quad & \int_T b(t)d\mu(t) \\ \text{subject to} \quad & \mu \in M_+(T), \quad \boldsymbol{c} - \int_T \boldsymbol{a}(t)d\mu(t) \in \mathcal{K}^n, \end{aligned} \tag{2.2}$$

where

$$\int_T \boldsymbol{a}(t)d\mu(t) := \left(\int_T a_1(t)d\mu(t), \dots, \int_T a_n(t)d\mu(t)\right)^\top \in \Re^n.$$

For any primal and dual feasible pair $(\overline{\boldsymbol{x}}, \overline{\mu})$, the weak duality holds, that is,

$$\boldsymbol{c}^\top \overline{\boldsymbol{x}} \geq \int_T b(t)d\overline{\mu}(t),$$

since $\boldsymbol{x}, \boldsymbol{y} \in \mathcal{K}^n$ entails $\boldsymbol{x}^\top \boldsymbol{y} \geq 0$. Moreover, under some mild assumptions, the strong duality also holds.

Proposition 2.1 (Strong duality theorem)

(a) *Suppose that the primal problem (1.2) has an optimum $\boldsymbol{x}^*$, and there exists a feasible point $\overline{\boldsymbol{x}} \in \mathcal{K}^n$ such that $\boldsymbol{a}(t)^\top \overline{\boldsymbol{x}} - b(t) > 0$ for any $t \in T$. Then the dual problem (2.2) has an optimum $\mu^* \in M_+(T)$ such that $\boldsymbol{c}^\top \boldsymbol{x}^* = \int_T b(t)d\mu^*(t)$.*

(b) *Suppose that the dual problem (2.2) has an optimum ν^*, and there exists a feasible point $\overline{\mu} \in M_+(T)$ such that $\boldsymbol{c} - \int_T \boldsymbol{a}(t)d\overline{\mu}(t) \in \operatorname{int}\mathcal{K}^n$ for any $t \in T$. Then the primal problem (1.2) has an optimum $\boldsymbol{x}^*$ such that $\boldsymbol{c}^\top \boldsymbol{x}^* = \int_T b(t)d\mu^*(t)$.*

By using the strong duality theorem, we can derive the complementarity slackness conditions for LSSIP (1.2). We omit the proof since it can be obtained in a trivial manner.

Proposition 2.2 (Complementarity slackness conditions) *Let $\boldsymbol{x}^* \in \Re^n$ and $\mu^* \in M(T)$ be feasible solutions for (1.2) and (2.2), respectively. Suppose that the assumption in either Proposition 2.1 (a) or (b) holds. Then, both $\boldsymbol{x}^*$ and μ^* are optimal if and only if*

$$\mu^* \in M_+(T), \ \boldsymbol{a}(t)^\top \boldsymbol{x}^* - b(t) \geq 0 \ (\forall t \in T), \ \int_T \left(\boldsymbol{a}(t)^\top \boldsymbol{x}^* - b(t) \right) d\mu^*(t) = 0,$$

$$\boldsymbol{x}^* \in \mathcal{K}^n, \ \boldsymbol{c} - \int_T \boldsymbol{a}(t) d\mu^*(t) \in \mathcal{K}^n, \ (\boldsymbol{x}^*)^\top \left\{ \boldsymbol{c} - \int_T \boldsymbol{a}(t) d\mu^*(t) \right\} = 0. \tag{2.3}$$

The vectors $\boldsymbol{x} \in \Re^n$ and $\boldsymbol{z} \in \Re^n$ are said to satisfy the *SOC complementarity conditions* if

$$\boldsymbol{x} \in \mathcal{K}^n, \ \boldsymbol{z} \in \mathcal{K}^n, \ \text{and} \ \boldsymbol{x}^\top \boldsymbol{z} = 0. \tag{2.4}$$

In (2.3), the vectors $\boldsymbol{x}^*$ and $\boldsymbol{c} - \int_T \boldsymbol{a}(t) d\mu^*(t)$ satisfy the SOC complementarity conditions.

If two vectors satisfy the SOC complementarity conditions, then the following proposition holds.

Proposition 2.3 *Suppose that n-dimensional vectors $\boldsymbol{x}$ and $\boldsymbol{z}$ satisfy the SOC complementarity conditions (2.4). Then, either of the following three cases holds for spectral values:*

(a) $0 = \lambda_1(\boldsymbol{x}) = \lambda_2(\boldsymbol{x})$ *and* $0 \leq \lambda_1(\boldsymbol{z}) \leq \lambda_2(\boldsymbol{z})$ *(i.e., $\boldsymbol{x} = \boldsymbol{0}$ and $\boldsymbol{z} \in \mathcal{K}^n$),*

(b) $0 \leq \lambda_1(\boldsymbol{x}) \leq \lambda_2(\boldsymbol{x})$ *and* $0 = \lambda_1(\boldsymbol{z}) = \lambda_2(\boldsymbol{z})$ *(i.e., $\boldsymbol{x} \in \mathcal{K}^n$ and $\boldsymbol{z} = \boldsymbol{0}$),*

(c) $0 = \lambda_1(\boldsymbol{x}) \leq \lambda_2(\boldsymbol{x})$ *and* $0 = \lambda_1(\boldsymbol{z}) \leq \lambda_2(\boldsymbol{z})$ *(i.e., $\boldsymbol{x} \in \mathrm{bd}\,\mathcal{K}^n$,*
$$\boldsymbol{z} \in \mathrm{bd}\,\mathcal{K}^n, \ \text{and} \ \boldsymbol{x} \perp \boldsymbol{z}).$$

Moreover, the spectral vectors for $\boldsymbol{x}$ and $\boldsymbol{z}$ can be chosen so that

$$\boldsymbol{v}_1(\boldsymbol{x}) = \boldsymbol{v}_2(\boldsymbol{z}) \ \text{and} \ \boldsymbol{v}_2(\boldsymbol{x}) = \boldsymbol{v}_1(\boldsymbol{z}). \tag{2.5}$$

Proof. Suppose that $\boldsymbol{x}$ and $\boldsymbol{z}$ satisfy the SOC complementarity conditions (2.4). It is obvious that (a) and (2.5) hold if $\boldsymbol{x} = \boldsymbol{0}$, and (b) and (2.5) also hold if $\boldsymbol{z} = \boldsymbol{0}$. Therefore, we only consider the case where $\boldsymbol{x} \neq \boldsymbol{0}$ and $\boldsymbol{z} \neq \boldsymbol{0}$. If $\tilde{\boldsymbol{x}} = \boldsymbol{0}$, then it follows $0 = \boldsymbol{x}^\top \boldsymbol{z} = x_1 z_1 + \tilde{\boldsymbol{x}}^\top \tilde{\boldsymbol{z}} = x_1 z_1$, i.e., $x_1 = 0$ or $z_1 = 0$. However, this together with $x_1 \geq \|\tilde{\boldsymbol{x}}\|$ and $z_1 \geq \|\tilde{\boldsymbol{z}}\|$ yields $\boldsymbol{x} = \boldsymbol{0}$ or $\boldsymbol{z} = \boldsymbol{0}$. Hence, we have $\tilde{\boldsymbol{x}} \neq \boldsymbol{0}$. We also have $\tilde{\boldsymbol{z}} \neq \boldsymbol{0}$ in an analogous way. Now, we first show (2.5). Let θ denote the angle between $\tilde{\boldsymbol{x}}$ and $\tilde{\boldsymbol{z}}$. Then we have $0 = x_1 z_1 + \tilde{\boldsymbol{x}}^\top \tilde{\boldsymbol{z}} \geq \|\tilde{\boldsymbol{x}}\|\|\tilde{\boldsymbol{z}}\| + \tilde{\boldsymbol{x}}^\top \tilde{\boldsymbol{z}} = \|\tilde{\boldsymbol{x}}\|\|\tilde{\boldsymbol{z}}\|(1 + \cos\theta)$. Since $\tilde{\boldsymbol{x}} \neq \boldsymbol{0}$ and $\tilde{\boldsymbol{z}} \neq \boldsymbol{0}$, we have $\cos\theta = -1$, which implies $\tilde{\boldsymbol{x}} = -\alpha\tilde{\boldsymbol{z}}$ for some $\alpha > 0$. We thus obtain (2.5). Next we show (c), i.e., $\lambda_1(\boldsymbol{x}) = \lambda_1(\boldsymbol{z}) = 0$. Note that $\cos\theta = -1$ implies $\tilde{\boldsymbol{x}}^\top \tilde{\boldsymbol{z}} = -\|\tilde{\boldsymbol{x}}\|\|\tilde{\boldsymbol{z}}\|$. Then we have $0 = x_1 z_1 + \tilde{\boldsymbol{x}}^\top \tilde{\boldsymbol{z}} = x_1 z_1 - \|\tilde{\boldsymbol{x}}\|\|\tilde{\boldsymbol{z}}\| \geq x_1 \|\tilde{\boldsymbol{z}}\| - \|\tilde{\boldsymbol{x}}\|\|\tilde{\boldsymbol{z}}\|$, which together with $\tilde{\boldsymbol{z}} \neq \boldsymbol{0}$ implies $x_1 \leq \|\tilde{\boldsymbol{x}}\|$. Since $x_1 \geq \|\tilde{\boldsymbol{x}}\|$ from $\boldsymbol{x} \in \mathcal{K}^n$, we have $x_1 - \|\tilde{\boldsymbol{x}}\| = \lambda_1(\boldsymbol{x}) = 0$. We also obtain $\lambda_1(\boldsymbol{z}) = 0$ in a similar manner. This completes the proof. ∎

From Proposition 2.3, the spectral factorizations for vectors $\boldsymbol{x}$ and $\boldsymbol{z}$ satisfying SOC complementarity conditions can be rewritten as

$$\boldsymbol{x} = \hat{x}_1 \hat{\boldsymbol{e}}_1 + \hat{x}_2 \hat{\boldsymbol{e}}_2, \quad \boldsymbol{z} = \hat{z}_1 \hat{\boldsymbol{e}}_1 + \hat{z}_2 \hat{\boldsymbol{e}}_2, \tag{2.6}$$

where

$$\hat{x}_1 = \lambda_1(\boldsymbol{x})/\sqrt{2}, \quad \hat{x}_2 = \lambda_2(\boldsymbol{x})/\sqrt{2}, \quad \hat{z}_1 = \lambda_2(\boldsymbol{z})/\sqrt{2}, \quad \hat{z}_2 = \lambda_1(\boldsymbol{z})/\sqrt{2}$$

$$\hat{\boldsymbol{e}}_1 = \sqrt{2}\boldsymbol{v}_1(\boldsymbol{x}) = \sqrt{2}\boldsymbol{v}_2(\boldsymbol{z}), \quad \hat{\boldsymbol{e}}_2 = \sqrt{2}\boldsymbol{v}_2(\boldsymbol{x}) = \sqrt{2}\boldsymbol{v}_1(\boldsymbol{z}).$$

Moreover, the following statements hold.

(a) $\hat{\boldsymbol{e}}_1, \hat{\boldsymbol{e}}_2 \in \operatorname{bd} \mathcal{K}^n$, $\|\hat{\boldsymbol{e}}_1\| = \|\hat{\boldsymbol{e}}_2\| = 1$, $(\hat{\boldsymbol{e}}_1)^\top \hat{\boldsymbol{e}}_2 = 0$, and $\hat{\boldsymbol{e}}_1 + \hat{\boldsymbol{e}}_2 = (\sqrt{2}, \boldsymbol{0})^\top$.

(b) $0 \le \hat{x}_1 \le \hat{x}_2$, $0 \le \hat{z}_2 \le \hat{z}_1$, and $\min\{\hat{x}_1, \hat{z}_1\} = \min\{\hat{x}_2, \hat{z}_2\} = 0$.

In the subsequent analyses, we adopt the expression (2.6) instead of the original spectral factorization.

The above statement (b) and the expression (2.6) correspond to the componentwise expression for the usual complementarity conditions and the orthonormal system in $\Re^n$, respectively. If the vectors $\boldsymbol{x}$ and $\boldsymbol{z}$ satisfy the usual complementarity conditions:

$$\boldsymbol{x} \ge \boldsymbol{0}, \ \boldsymbol{z} \ge \boldsymbol{0}, \ \text{and} \ \boldsymbol{x}^\top \boldsymbol{z} = 0,$$

then it follows for every i, $x_i \ge 0$, $z_i \ge 0$, and $\min\{x_i, z_i\} = 0$. In this case, $\boldsymbol{x}$ and $\boldsymbol{z}$ can be decomposed as $\boldsymbol{x} = \sum_{i=1}^n x_i \boldsymbol{e}_i$ and $\boldsymbol{z} = \sum_{i=1}^n z_i \boldsymbol{e}_i$, respectively, where $\boldsymbol{e}_i$ is the n-dimensional vector whose i-th component is 1 and other components are 0. Similarly, $\hat{\boldsymbol{e}}_1$ and $\hat{\boldsymbol{e}}_2$ are the orthonormal bases for the 2-dimensional subspace containing $\boldsymbol{x}$, $\boldsymbol{z}$, and the axis[2] of the SOC. We note, however, that $\hat{\boldsymbol{e}}_1$ and $\hat{\boldsymbol{e}}_2$ depend on $(\boldsymbol{x}, \boldsymbol{z})$, while $\boldsymbol{e}_i$ is fixed for every i.

3 Explicit exchange algorithm

In this section, we propose an explicit exchange algorithm for solving LSSIP (1.2). Moreover, we show that the algorithm terminates in a finite number of iterations, and the obtained solution is sufficiently close to the original LSSIP solution if the criterion value is sufficiently close to 0.

In the algorithm, we solve the finitely constrained SOCP as the subproblem for each iteration. For a finite set $T' := \{t_1, \ldots, t_m\} \subset T$, we define the finitely constrained linear SOCP, say LSOCP(T'), as

$$\begin{aligned}
\text{Minimize} \quad & \boldsymbol{c}^\top \boldsymbol{x} \\
\text{subject to} \quad & \boldsymbol{x} \in \mathcal{K}^n, \quad \boldsymbol{a}(t_j)^\top \boldsymbol{x} - b(t_j) \ge 0 \quad (j = 1, \ldots, m),
\end{aligned}$$

and its dual problem DLSOCP(T') as

$$\begin{aligned}
\text{Maximize} \quad & \sum_{j=1}^m b(t_j) \nu(t_j) \\
\text{subject to} \quad & \nu(t_j) \ge 0, \quad \boldsymbol{c} - \sum_{j=1}^m \boldsymbol{a}(t_j) \nu(t_j) \in \mathcal{K}^n.
\end{aligned}$$

Here, the dual variable $\{\nu(t_j)\}_{j=1}^m$ can be regarded as an m-dimensional real vector. In the remainder of the paper, we denote the optimal values of LSOCP(T') and DLSOCP(T') by $V_P(T')$ and $V_D(T')$, respectively. Moreover, we denote the optimal value of LSSIP (1.2) by $V_P(T)$ for convenience.

One possible choice for solving LSSIP (1.2) is to apply the discretization method, which solves LSOCP(T^k) at each iteration and updates $\{T^k\}$ so that $\operatorname{dist}(T^k, T) \to 0$. Another possible choice is the implicit cutting plane algorithm, which solves LSOCP(T^k) at each iteration and updates $\{T^k\}$ so that $T^{k+1} := T^k \cup \{t^k\}$ with $t^k \in \operatorname{argmin}_{t \in T}\{\boldsymbol{a}(t)^\top \boldsymbol{x}^k - b(t)\}$. These two algorithms are intuitively comprehensible, and the convergence analyses in the existing literature may be applied to SOC version directly. However, they have fatal flaws. As k becomes larger, the constraints of each subproblem

---

[2]The axis of SOC is defined by $\{(\alpha, 0, \ldots, 0)^\top \in \Re^n \mid \alpha \in \Re\}$.

increase unboundedly, and hence it becomes more and more expensive. In order to avoid such a difficulty, we propose an explicit exchange algorithm based on the approach in [18]. The algorithm keeps the size of subproblems bounded by removing inactive constraints. Moreover, the global minimization problem with respect to $t \in T$ need not be solved in each iteration. The detailed steps are given as follows.

Algorithm 3.1

> **Step 0** *Choose a small real number $\gamma > 0$ and finite points $E^0 := \{t_1^0, \ldots, t_{m_0}^0\} \subset T$. Solve $\mathrm{LSOCP}(E^0)$ to obtain its optimum $\boldsymbol{x}^0$. Set $k := 0$.*
>
> **Step 1** *Find a $t_{\mathrm{new}}^k \in T$ such that*
>
> $$\boldsymbol{a}(t_{\mathrm{new}}^k)^\top \boldsymbol{x}^k - b(t_{\mathrm{new}}^k) < -\gamma. \tag{3.1}$$
>
> *If such a t_{new}^k does not exist, i.e., $\min_{t \in T}\{\boldsymbol{a}(t)^\top \boldsymbol{x}^k - b(t)\} \geq -\gamma$, then terminate. Otherwise, let*
>
> $$\overline{E}^{k+1} := E^k \cup \{t_{\mathrm{new}}^k\}.$$
>
> **Step 2** *Solve $\mathrm{LSOCP}(\overline{E}^{k+1})$ and $\mathrm{DLSOCP}(\overline{E}^{k+1})$ to obtain their solutions $\boldsymbol{x}^{k+1}$ and $\{\nu_{k+1}(t) \mid t \in \overline{E}^{k+1}\}$, respectively.*
>
> **Step 3** *Let*
>
> $$E^{k+1} := \{t \in \overline{E}^{k+1} \mid \nu_{k+1}(t) > 0\}.$$
>
> *Set $k := k + 1$ and return to Step 1.*

In Step 1, it is also possible to choose multiple elements satisfying (3.1). Although we only deal with the one-point exchange scheme in the following analyses, they are also applicable to multiple exchange type algorithms. In Step 2, $\mathrm{LSOCP}(\overline{E}^{k+1})$ and $\mathrm{DLSOCP}(\overline{E}^{k+1})$ can be solved simultaneously by using an existing method. Here, we note that $\boldsymbol{x}^{k+1}$ also solves $\mathrm{LSOCP}(E^{k+1})$. In Step 3, all inactive constraints at the optimum $\boldsymbol{x}^k$ are removed, since the complementarity conditions imply that $\boldsymbol{a}(t)^\top \boldsymbol{x} - b(t) = 0$ for any t such that $\nu_{k+1}(t) > 0$. This step is the main difference from the discretization method and the implicit cutting plane algorithm.

Before showing the finite termination of the algorithm, we define some notations for convenience. Let $\{\boldsymbol{x}^k\}$ and $\{\nu_k\}$ be the sequence generated by Algorithm 3.1. Then, we denote

$$Z^k := \{t \in \overline{E}^k \mid \nu_k(t) = 0\}, \; y^k(t) := \boldsymbol{a}(t)^\top \boldsymbol{x}^k - b(t), \; \boldsymbol{z}^k := \boldsymbol{c} - \sum_{t \in E^k} \boldsymbol{a}(t)\nu_k(t).$$

It is obvious that $E^k \cup Z^k = \overline{E}^k$ and $E^k \cap Z^k = \emptyset$. Moreover, from the complementarity slackness conditions, it follows

$$\nu_k(t) \geq 0, \quad y^k(t) \geq 0, \quad \nu^k(t)\, y^k(t) = 0, \qquad (\forall t \in \overline{E}^k)$$
$$\boldsymbol{x}^k \in \mathcal{K}^n, \quad \boldsymbol{z}^k \in \mathcal{K}^n, \quad (\boldsymbol{x}^k)^\top \boldsymbol{z}^k = 0.$$

We decompose the iteration points $\boldsymbol{x}^k$ and $\boldsymbol{z}^k$ in a way analogous to (2.6) as follows:

$$\boldsymbol{x}^k = \hat{x}_1^k \hat{\boldsymbol{e}}_1^k + \hat{x}_2^k \hat{\boldsymbol{e}}_2^k \quad \text{and} \quad \boldsymbol{z}^k = \hat{z}_1^k \hat{\boldsymbol{e}}_1^k + \hat{z}_2^k \hat{\boldsymbol{e}}_2^k.$$

Notice that $(\hat{\boldsymbol{e}}_1^k, \hat{\boldsymbol{e}}_2^k) \neq (\hat{\boldsymbol{e}}_1^{k+1}, \hat{\boldsymbol{e}}_2^{k+1})$ in general. We further define the indices sets by

$$U_+^k := \{i \in \{1,2\} \,|\, \hat{x}_i^k > 0\}, \quad U_0^k := \{i \in \{1,2\} \,|\, \hat{x}_i^k = 0\}.$$

For $i = 1$ and 2, $i \in U_+^k$ implies $\hat{z}_i^k = 0$ since $\min\{\hat{x}_i^k, \hat{z}_i^k\} = 0$. Moreover, due to $\hat{x}_1^k \leq \hat{x}_2^k$, only the following three cases are possible:

(i) $U_0^k = \emptyset$, $U_+^k = \{1,2\}$, (ii) $U_0^k = \{1\}$, $U_+^k = \{2\}$, (iii) $U_0^k = \{1,2\}$, $U_+^k = \emptyset$.

Next, we give some assumptions and lemmas that are necessary for the subsequent analyses. We first introduce the following assumption.

Assumption A *For all k, (i) $\mathrm{LSOCP}(\overline{E}^k)$ and $\mathrm{DLSOCP}(\overline{E}^k)$ have unique solutions respectively, and (ii) $V_P(\overline{E}^k) = V_D(\overline{E}^k)$.*

Assumption A(i) usually holds since the feasible regions of $\mathrm{LSOCP}(\overline{E}^k)$ and $\mathrm{DLSOCP}(\overline{E}^k)$ are intersections of a polyhedral and an SOC (or its affine transformation), respectively. Assumption A(ii) is the strong duality, which also holds in most cases. Under the above assumption, we have the following lemma.

Lemma 3.1 *Suppose that Assumption A holds. Then, it follows for all $k \geq 0$ that*

(a) $V_P(E^k) < V_P(E^{k+1})$,

(b) $t_{\mathrm{new}}^k \in E^{k+1}$.

Proof. We first show (a). Let $\mathcal{F}^k$ and $\overline{\mathcal{F}}^k$ be the feasible regions of $\mathrm{LSOCP}(E^k)$ and $\mathrm{LSOCP}(\overline{E}^k)$, respectively. Then, we have $\boldsymbol{x}^k \neq \boldsymbol{x}^{k+1}$ since $\boldsymbol{x}^{k+1} \in \overline{\mathcal{F}}^{k+1}$ and $\boldsymbol{x}^k \notin \overline{\mathcal{F}}^{k+1}$. Moreover, $V_P(E^k) \leq V_P(E^{k+1})$ since $V_P(\overline{E}^{k+1}) = V_P(E^{k+1})$ and $\mathcal{F}^k \supseteq \overline{\mathcal{F}}^{k+1}$. Now, we assume $V_P(E^k) = V_P(E^{k+1})$ for contradiction. Then, $\boldsymbol{x}^{k+1}$ solves not only $\mathrm{LSOCP}(\overline{E}^{k+1})$ but also $\mathrm{LSOCP}(E^k)$ since $V_P(E^k) = V_P(E^{k+1}) = V_P(\overline{E}^{k+1})$ and $\mathcal{F}^k \supseteq \overline{\mathcal{F}}^{k+1}$. Hence, both $\boldsymbol{x}^k$ and $\boldsymbol{x}^{k+1}$ solve $\mathrm{LSOCP}(\overline{E}^k)$. However, we have $\boldsymbol{x}^k \neq \boldsymbol{x}^{k+1}$, which contradicts Assumption A. Hence we have (a).

Next we show (b). Assume $t_{\mathrm{new}}^k \notin E^{k+1}$ for contradiction. Then we have $\nu_{k+1}(t_{\mathrm{new}}^k) = 0$, which yields $V_D(\overline{E}^{k+1}) = b(t_{\mathrm{new}}^k)\nu_{k+1}(t_{\mathrm{new}}^k) + \sum_{t \in E^k} b(t)\nu_{k+1}(t) = 0 + \sum_{t \in E^k} b(t)\nu_{k+1}(t) = V_D(E^k)$. This, together with $V_D(\overline{E}^{k+1}) = V_P(E^{k+1})$ and $V_D(E^k) = V_P(E^k)$, implies $V_P(E^{k+1}) = V_P(E^k)$, which contradicts (a). We thus have (b). ∎

The following lemma evaluates the increment of $\{V_P(E^k)\}$ in each iteration.

Lemma 3.2 *Suppose that Assumption A holds. Then we have*

$$V_P(E^{k+1}) - V_P(E^k) = \sum_{t \in Z^{k+1}} y^{k+1}(t)\nu_k(t) + (\boldsymbol{z}^k)^\top \boldsymbol{x}^{k+1}$$
$$= -(\boldsymbol{z}^{k+1})^\top \boldsymbol{x}^k - y^k(t_{\mathrm{new}}^k)\nu_{k+1}(t_{\mathrm{new}}^k).$$

Proof. From the definition of $\boldsymbol{z}^k$, we have

$$(\boldsymbol{z}^k)^\top \boldsymbol{x}^{k+1} = \left\{ \boldsymbol{c} - \sum_{t \in E^k} \boldsymbol{a}(t)\nu_k(t) \right\}^\top \boldsymbol{x}^{k+1}$$

$$= -\sum_{t \in E^k} \boldsymbol{a}(t)^\top \boldsymbol{x}^{k+1} \nu_k(t) + \boldsymbol{c}^\top \boldsymbol{x}^{k+1}$$

$$= -\sum_{t \in E^k} \left(y^{k+1}(t) + b(t) \right) \nu_k(t) + V_P(E^{k+1})$$

$$= -\sum_{t \in E^k} y^{k+1}(t)\nu_k(t) - \sum_{t \in E^k} b(t)\nu_k(t) + V_P(E^{k+1})$$

$$= -\sum_{t \in Z^{k+1}} y^{k+1}(t)\nu_k(t) - V_P(E^k) + V_P(E^{k+1}),$$

where the last equality holds since

$$\sum_{t \in E^k} b(t)\nu_k(t) = \sum_{t \in \overline{E}^k} b(t)\nu_k(t) = V_D(\overline{E}^k) = V_P(E^k)$$

and

$$\sum_{t \in E^k} y^{k+1}(t)\nu_k(t) = -y^{k+1}(t_{\mathrm{new}}^k)\nu_k(t_{\mathrm{new}}^k) + \sum_{t \in \overline{E}^{k+1}} y^{k+1}(t)\nu_k(t)$$

$$= 0 + \sum_{t \in Z^{k+1}} y^{k+1}(t)\nu_k(t)$$

with $y^{k+1}(t) = 0$ for $t \in E^{k+1}$ and $t_{\mathrm{new}}^k \in E^{k+1}$. Hence, we have the first equality.

Next, we show the second equality. From the definition of $y^k(t)$, we have

$$\sum_{t \in \overline{E}^{k+1}} y^k(t)\nu_{k+1}(t) = \sum_{t \in \overline{E}^{k+1}} \left(\boldsymbol{a}(t)^\top \boldsymbol{x}^k - b(t) \right) \nu_{k+1}(t)$$

$$= \left(\sum_{t \in \overline{E}^{k+1}} \boldsymbol{a}(t)\nu_{k+1}(t) \right)^\top \boldsymbol{x}^k - \sum_{t \in \overline{E}^{k+1}} b(t)\nu_{k+1}(t)$$

$$= \left(\sum_{t \in E^{k+1}} \boldsymbol{a}(t)\nu_{k+1}(t) \right)^\top \boldsymbol{x}^k - \sum_{t \in E^{k+1}} b(t)\nu_{k+1}(t)$$

$$= (-\boldsymbol{z}^{k+1} + \boldsymbol{c})^\top \boldsymbol{x}^k - V_P(E^{k+1})$$

$$= -(\boldsymbol{z}^{k+1})^\top \boldsymbol{x}^k + V_P(E^k) - V_P(E^{k+1}).$$

Since $y^k(t) = 0$ for all $t \in E^k$, the above equality yields

$$V_P(E^{k+1}) - V_P(E^k) = -(\boldsymbol{z}^{k+1})^\top \boldsymbol{x}^k - y^k(t_{\mathrm{new}}^k)\nu_{k+1}(t_{\mathrm{new}}^k).$$

This completes the proof. ∎

Next, we introduce a second group of assumptions.

Assumption B *There exist $M > 0$ and $\delta > 0$ such that the following statements hold for all $k \geq 1$.*

    (a) $\|\boldsymbol{x}^k\| \leq M$, $\|\boldsymbol{z}^k\| \leq M$.

    (b) $\nu_k(t) \geq \delta$ *for all* $t \in E^k$.

    (c) $\delta \leq \max\{\hat{x}_i^k, \hat{z}_i^k\} \leq M$ *for* $i = 1, 2$.

    (d)   *If* $\boldsymbol{x}^k \in \mathrm{bd}\,\mathcal{K}^n$ *and* $\boldsymbol{x}^k \neq \boldsymbol{0}$, *then there exists* $t \in E^k$ *such that* $\boldsymbol{a}(t)^\top \hat{\boldsymbol{e}}_2^k \notin (-\delta, 0]$.

(e) *If $\boldsymbol{x}^k \in \operatorname{int} \mathcal{K}^n$, then it follows $\lambda_{\min}(H_k H_k^\top) \geq \delta$, where $\lambda_{\min}$ denotes the minimum eigenvalue, and the matrix H_k is defined by*

$$H_k := \left(\boldsymbol{a}(t_1^k), \ldots, \boldsymbol{a}(t_{m_k}^k) \right) \in \Re^{n \times m_k} \tag{3.2}$$

with $E^k = \{t_1^k, \ldots, t_{m_k}^k\}$.

Assumption B (a) asserts the boundedness of the generated sequence. Assumptions B (b) and (c) say that the complementarity slackness conditions for each subproblem holds strictly enough. Moreover, Assumption B (c) together with $\min\{\hat{x}_i^k, \hat{z}_i^k\} = 0$ implies

$$i \in U_0^k \iff \hat{x}_i^k = 0, \ \hat{z}_i^k \geq \delta,$$
$$i \in U_+^k \iff \hat{x}_i^k \geq \delta, \ \hat{z}_i^k = 0.$$

Assumptions B (d) and (e) claim a kind of regularity, which holds in most cases. Indeed, Assumption B (d) is violated only when $\boldsymbol{a}(t)^\top \hat{\boldsymbol{e}}_2^k$ belongs to the minute interval $(-\delta, 0]$ for all $t \in E^k$. Assumption B (e) implies that the matrix H_k is sufficiently full rank. We note, however, that the vector $\boldsymbol{x}^k$ is usually on the boundary of $\mathcal{K}^n$. Otherwise, the SOC constraint in LSOCP $(\overline{E}^k)$ exercises no influence on the solution $\boldsymbol{x}^k$.

By using the aforementioned lemmas, we derive the main theorem of this section.

Theorem 3.1 *Suppose that Assumptions A and B hold. Then, Algorithm 3.1 terminates in a finite number of iterations.*

Proof. Assume that Algorithm 3.1 does not terminate in finite iterations for contradiction. Then, from Lemma 3.1 (a) and $V_P(E^k) \leq V_P(T)$, we have $\lim_{k \to \infty}(V_P(E^{k+1}) - V_P(E^k)) = 0$. Moreover, Assumption B(a) and $\{t_{\text{new}}^k\} \subseteq T$ imply the existence of $\boldsymbol{x}^a \in \Re^n$, $t^a \in T$ and $S \subseteq \{0, 1, 2, \ldots\}$ such that $\lim_{k \to \infty, k \in S}(\boldsymbol{x}^k, t_{\text{new}}^k) = (\boldsymbol{x}^a, t^a)$. Let $y^a : T \to \Re$ be defined by

$$y^a(t) := \boldsymbol{a}(t)^\top \boldsymbol{x}^a - b(t).$$

Then, we have $\lim_{k \to \infty, k \in S} y^k(t_{\text{new}}^k) = y^a(t^a)$. Since $y^k(t_{\text{new}}^k) < -\gamma$ for all k, it follows

$$y^a(t^a) \leq -\gamma. \tag{3.3}$$

Now, choose any sufficiently small ε such that $0 < \varepsilon < \min(\delta, \delta^2)$. Then, we can find a large integer $N = N(\varepsilon) \in S$ such that

$$0 < V_P(E^{N+1}) - V_P(E^N) < \varepsilon^2, \tag{3.4}$$
$$\left| y^a(t^a) - y^N(t_{\text{new}}^N) \right| < \varepsilon^2. \tag{3.5}$$

In what follows, we divide the proof in two parts. In Part 1, we show

$$U_+^N = U_+^{N+1} \quad \text{and} \quad U_0^N = U_0^{N+1}. \tag{3.6}$$

By using this result, we derive a contradiction in Part 2.

(Part 1) Since $U_+^N \cup U_0^N = \{1, 2\}$ and $U_+^N \cap U_0^N = \emptyset$, it suffices to show $U_0^N \subseteq U_0^{N+1}$ and $U_+^N \subseteq U_+^{N+1}$. From (3.4) and Lemma 3.2, we have

$$0 < \sum_{t \in Z^{N+1}} y^{N+1}(t)\nu_N(t) + (\boldsymbol{z}^N)^\top \boldsymbol{x}^{N+1} < \varepsilon^2. \tag{3.7}$$

From $\boldsymbol{z}^N \in \mathcal{K}^n$ and $\boldsymbol{x}^{N+1} \in \mathcal{K}^n$ we have $(\boldsymbol{z}^N)^\top \boldsymbol{x}^{N+1} \geq 0$, which together with (3.7) and $\sum_{t \in Z^{N+1}}$ $y^{N+1}(t)\nu_N(t) \geq 0$ yields $(\boldsymbol{z}^N)^\top \boldsymbol{x}^{N+1} < \varepsilon^2$. Hence, we have

$$
\begin{aligned}
\varepsilon^2 &> (\boldsymbol{z}^N)^\top \boldsymbol{x}^{N+1} \\
&= (\hat{z}_1^N \hat{\boldsymbol{e}}_1^N + \hat{z}_2^N \hat{\boldsymbol{e}}_2^N)^\top (\hat{x}_1^{N+1} \hat{\boldsymbol{e}}_1^{N+1} + \hat{x}_2^{N+1} \hat{\boldsymbol{e}}_2^{N+1}).
\end{aligned} \tag{3.8}
$$

First we show $U_0^N \subseteq U_0^{N+1}$. If $U_0^N = \{1\}$, then $\hat{z}_2^N = 0$. Hence, (3.8) reduces to

$$
\begin{aligned}
\varepsilon^2 &> (\hat{z}_1^N \hat{\boldsymbol{e}}_1^N)^\top (\hat{x}_1^{N+1} \hat{\boldsymbol{e}}_1^{N+1} + \hat{x}_2^{N+1} \hat{\boldsymbol{e}}_2^{N+1}) \\
&= (\hat{z}_1^N \hat{\boldsymbol{e}}_1^N)^\top \left\{ \hat{x}_1^{N+1}(\hat{\boldsymbol{e}}_1^{N+1} + \hat{\boldsymbol{e}}_2^{N+1}) \right\} + (\hat{z}_1^N \hat{\boldsymbol{e}}_1^N)^\top (\hat{x}_2^{N+1} - \hat{x}_1^{N+1}) \hat{\boldsymbol{e}}_2^{N+1} \\
&\geq \hat{z}_1^N \hat{x}_1^{N+1} (\hat{\boldsymbol{e}}_1^N)^\top (\hat{\boldsymbol{e}}_1^{N+1} + \hat{\boldsymbol{e}}_2^{N+1}) \\
&= \hat{z}_1^N \hat{x}_1^{N+1},
\end{aligned}
$$

where the second inequality is due to $\hat{x}_1^{N+1} \leq \hat{x}_2^{N+1}$ and $(\hat{\boldsymbol{e}}_1^N)^\top \hat{\boldsymbol{e}}_2^{N+1} \geq 0$, and the last equality holds from $\hat{\boldsymbol{e}}_1^{N+1} + \hat{\boldsymbol{e}}_2^{N+1} = \hat{\boldsymbol{e}}_1^N + \hat{\boldsymbol{e}}_2^N = (\sqrt{2}, \boldsymbol{0})^\top$ and $(\hat{\boldsymbol{e}}_i^N)^\top (\hat{\boldsymbol{e}}_j^N) = \delta_{ij}$. Thus we have $\hat{x}_1^{N+1} < \varepsilon^2/\hat{z}_1^N \leq \varepsilon^2/\delta < \varepsilon < \delta$, which implies $\hat{x}_1^{N+1} = 0$ from Assumption B (c). If $U_0^N = \{1, 2\}$, then $\delta \leq \hat{z}_2^N \leq \hat{z}_1^N$, and hence we have from (3.8) that

$$
\begin{aligned}
\varepsilon^2 &> (\hat{z}_1^N \hat{\boldsymbol{e}}_1^N + \hat{z}_2^N \hat{\boldsymbol{e}}_2^N)^\top (\hat{x}_1^{N+1} \hat{\boldsymbol{e}}_1^{N+1} + \hat{x}_2^{N+1} \hat{\boldsymbol{e}}_2^{N+1}) \\
&\geq (\hat{z}_1^N \hat{\boldsymbol{e}}_1^N + \hat{z}_2^N \hat{\boldsymbol{e}}_2^N)^\top (\hat{x}_2^{N+1} \hat{\boldsymbol{e}}_2^{N+1}) \\
&= \hat{z}_2^N (\hat{\boldsymbol{e}}_1^N + \hat{\boldsymbol{e}}_2^N)^\top (\hat{x}_2^{N+1} \hat{\boldsymbol{e}}_2^{N+1}) + \left\{ (\hat{z}_1^N - \hat{z}_2^N) \hat{\boldsymbol{e}}_1^N \right\}^\top (\hat{x}_2^{N+1} \hat{\boldsymbol{e}}_2^{N+1}) \\
&\geq \hat{z}_2^N \hat{x}_2^{N+1} (\hat{\boldsymbol{e}}_1^N + \hat{\boldsymbol{e}}_2^N)^\top \hat{\boldsymbol{e}}_2^{N+1} \\
&= \hat{z}_2^N \hat{x}_2^{N+1},
\end{aligned}
$$

where the second inequality follows since $(\hat{z}_1^N \hat{\boldsymbol{e}}_1^N + \hat{z}_2^N \hat{\boldsymbol{e}}_2^N)^\top (\hat{x}_i^{N+1} \hat{\boldsymbol{e}}_i^{N+1}) \geq 0$ for $i = 1$ and 2, the third inequality follows from $\hat{z}_1^N \geq \hat{z}_2^N$, and the last equality holds from $\hat{\boldsymbol{e}}_1^N + \hat{\boldsymbol{e}}_2^N = \hat{\boldsymbol{e}}_1^{N+1} + \hat{\boldsymbol{e}}_2^{N+1} = (\sqrt{2}, \boldsymbol{0})^\top$ and $(\hat{\boldsymbol{e}}_i^{N+1})^\top (\hat{\boldsymbol{e}}_j^{N+1}) = \delta_{ij}$. Hence, we have $\hat{x}_2^{N+1} < \varepsilon^2/\hat{z}_2^N \leq \varepsilon^2/\delta < \varepsilon < \delta$, which implies $\hat{x}_2^{N+1} = 0$ from Assumption B (c). Since $\hat{x}_1^{N+1} \leq \hat{x}_2^{N+1}$, we also have $\hat{x}_1^{N+1} = 0$.

Next we show $U_+^N \subseteq U_+^{N+1}$. Since $y^{N+1}(t) = 0$ for any $t \in E^{N+1}$ and $t_{\text{new}}^N \in E^{N+1}$, we have

$$
\sum_{t \in Z^{N+1}} y^{N+1}(t)\nu_N(t) = \sum_{t \in \overline{E}^{N+1}} y^{N+1}(t)\nu_N(t) = \sum_{t \in E^N} y^{N+1}(t)\nu_N(t).
$$

Hence, by (3.7), we have

$$
0 < \sum_{t \in E^N} y^{N+1}(t)\nu_N(t) + (\boldsymbol{z}^N)^\top \boldsymbol{x}^{N+1} < \varepsilon^2. \tag{3.9}
$$

Now, choose $t^N \in E^N$ arbitrarily. Since $(\boldsymbol{z}^N)^\top \boldsymbol{x}^{N+1} \geq 0$, $y^{N+1}(t^N) \geq 0$, and $\nu_N(t^N) > 0$, we have from (3.9) that $0 \leq y^{N+1}(t^N)\nu_N(t^N) < \varepsilon^2$. Dividing this by $\nu_N(t^N) \geq \delta$, we obtain $0 \leq y^{N+1}(t^N) < \varepsilon^2/\delta < \varepsilon$, which also implies

$$
0 \leq \boldsymbol{a}(t^N)^\top (\boldsymbol{x}^{N+1} - \boldsymbol{x}^N) < \varepsilon \tag{3.10}
$$

since $\boldsymbol{a}(t^N)^\top (\boldsymbol{x}^{N+1} - \boldsymbol{x}^N) = y^{N+1}(t^N) - y^N(t^N)$ and $y^N(t^N) = 0$. First, we consider the case where $U_+^N = \{2\}$, i.e., $\hat{x}_1^N = 0$ and $\hat{x}_2^N \geq \delta$. Assume that $2 \notin U_+^{N+1}$ for contradiction. Then, $\boldsymbol{x}^{N+1} = \boldsymbol{0}$ since $\hat{x}_2^{N+1} = 0$ and $0 \leq \hat{x}_1^{N+1} \leq \hat{x}_2^{N+1}$. Hence, (3.10) yields $0 \leq -\boldsymbol{a}(t^N)^\top (\hat{x}_2^N \hat{\boldsymbol{e}}_2^N) \leq \varepsilon$, that is,

$0 \leq -\boldsymbol{a}(t^N)^\top \hat{\boldsymbol{e}}_2^N \leq \varepsilon/\hat{x}_2^N < \delta^2/\delta = \delta$. However, this contradicts Assumption B (d) since t^N is an arbitrary element in E^N. Hence, we have $2 \in U_+^{N+1}$. Next, we consider the case where $U_+^N = \{1,2\}$, i.e., $\delta \leq \hat{x}_1^N \leq \hat{x}_2^N$. By (3.10), we have

$$H_N^\top(\boldsymbol{x}^{N+1} - \boldsymbol{x}^N) = O(\varepsilon),$$

where H_N is the matrix defined by (3.2) for $k := N$. By Assumption B (e), we have $\boldsymbol{x}^{N+1} - \boldsymbol{x}^N = O(\varepsilon)$, which implies $\hat{x}_i^{N+1} - \hat{x}_i^N = O(\varepsilon)$ for $i = 1, 2$ since $\sqrt{2}\hat{x}_i^N = x_1^N + (-1)^i\{(x_2^N)^2 + \cdots + (x_n^N)^2\}^{1/2}$ for $\boldsymbol{x}^N = (x_1^N, x_2^N, \ldots, x_n^N)^\top$. If $i \in U_+^N$, then $\hat{x}_i^N \geq \delta$, which together with $\hat{x}_i^{N+1} - \hat{x}_i^N = O(\varepsilon)$ yields $\hat{x}_i^{N+1} > 0$ (i.e., $\hat{x}_i^{N+1} \geq \delta$). Thus we obtain $U_+^N \subseteq U_+^{N+1}$.

(Part 2) From Lemma 3.2 and (3.4), we have

$$0 < \sum_{t \in Z^{N+1}} y^{N+1}(t)\nu_N(t) + (\boldsymbol{z}^N)^\top \boldsymbol{x}^{N+1} < \varepsilon^2, \tag{3.11}$$

$$0 < -(\boldsymbol{z}^{N+1})^\top \boldsymbol{x}^N - y^N(t_{\text{new}}^N)\nu_{N+1}(t_{\text{new}}^N) < \varepsilon^2. \tag{3.12}$$

Consider

$$(\boldsymbol{z}^{N+1})^\top \boldsymbol{x}^N = (\hat{z}_1^{N+1}\hat{\boldsymbol{e}}_1^{N+1} + \hat{z}_2^{N+1}\hat{\boldsymbol{e}}_2^{N+1})^\top(\hat{x}_1^N\hat{\boldsymbol{e}}_1^N + \hat{x}_2^N\hat{\boldsymbol{e}}_2^N).$$

Then, due to (3.6), the following three cases are possible: (i) $U_+^N = U_+^{N+1} = \emptyset$, (ii) $U_+^N = U_+^{N+1} = \{1,2\}$, (iii) $U_+^N = U_+^{N+1} = \{2\}$. If (i) holds, then $\hat{x}_1^N = \hat{x}_2^N = 0$, and hence $(\boldsymbol{z}^{N+1})^\top \boldsymbol{x}^N = 0$. If (ii) holds, then $\hat{z}_1^{N+1} = \hat{z}_2^{N+1} = 0$, and hence, $(\boldsymbol{z}^{N+1})^\top \boldsymbol{x}^N = 0$. If case (iii) holds, then $\hat{z}_2^N = \hat{x}_1^{N+1} = 0$, which together with (3.11) and $y^{N+1}(t)\nu_N(t) \geq 0$ for all $t \in Z^{N+1}$ yields

$$\varepsilon^2 > (\boldsymbol{z}^N)^\top \boldsymbol{x}^{N+1} = \hat{z}_1^N \hat{x}_2^{N+1}(\hat{\boldsymbol{e}}_1^N)^\top\hat{\boldsymbol{e}}_2^{N+1} \geq \delta^2(\hat{\boldsymbol{e}}_1^N)^\top\hat{\boldsymbol{e}}_2^{N+1}.$$

Since $(\hat{\boldsymbol{e}}_1^N)^\top\hat{\boldsymbol{e}}_2^{N+1} = (\hat{\boldsymbol{e}}_2^N)^\top\hat{\boldsymbol{e}}_1^{N+1}$ from the definition, it follows $(\hat{\boldsymbol{e}}_2^N)^\top\hat{\boldsymbol{e}}_1^{N+1} < \varepsilon^2/\delta^2$. We thus have

$$(\boldsymbol{z}^{N+1})^\top \boldsymbol{x}^N = \hat{z}_1^{N+1}\hat{x}_2^N(\hat{\boldsymbol{e}}_1^{N+1})^\top\hat{\boldsymbol{e}}_2^N < \frac{M^2\varepsilon^2}{\delta^2}$$

for every case (i) – (iii). This together with (3.12) yields $0 < -y^N(t_{\text{new}}^N)\nu_{N+1}(t_{\text{new}}^N) < (1 + M^2/\delta^2)\varepsilon^2$. Hence, by (3.4), we have

$$|y^a(t^a)| < |y^N(t_{\text{new}}^N)| + \varepsilon^2 < \frac{(1 + M^2/\delta^2)\varepsilon^2}{\nu_{N+1}(t_{\text{new}}^N)} + \varepsilon^2 \leq \left(1 + \frac{\delta^2 + M^2}{\delta^3}\right)\varepsilon^2,$$

where the last inequality follows since Lemma 3.1(b) and Assumption B(b) imply $\nu_{N+1}(t_{\text{new}}^N) \geq \delta$. Since $\varepsilon > 0$ can be chosen arbitrarily small, it must hold $y^a(t^a) = 0$, which contradicts (3.3). ∎

Thus far, we have shown the finite termination of Algorithm 3.1. Nevertheless, the above theorem would be meaningless if the obtained solution were far from the original LSSIP optimum. The following theorem guarantees that, if $\gamma > 0$ is sufficiently small, then the solution obtained by Algorithm 3.1 is also close to the original LSSIP optimum.

Theorem 3.2 *Suppose that Assumptions A and B hold. Let $\overline{k}(\gamma)$ be the number of iterations in which Algorithm 3.1 terminates. Then, it follows $\lim_{\gamma \to 0} V_P(E^{\overline{k}(\gamma)}) = V_P(T)$. Moreover, if LSSIP (1.2) has a unique solution $\boldsymbol{x}^*$, then $\lim_{\gamma \to 0} \boldsymbol{x}^{\overline{k}(\gamma)} = \boldsymbol{x}^*$.*

Proof. Let $\mathcal{F}$ be the feasible region of LSSIP (1.2), and $\mathcal{F}_\gamma$ be defined by

$$\mathcal{F}_\gamma := \left\{ \boldsymbol{x} \,\middle|\, \boldsymbol{x} \in \mathcal{K}^n,\ \boldsymbol{a}(t)^\top \boldsymbol{x} - b(t) \geq -\gamma \ (\forall t \in T) \right\}.$$

Then, we can see $\boldsymbol{x}^{\overline{k}(\gamma)} \in \mathcal{F}_\gamma$, $\mathcal{F}_\gamma \supseteq \mathcal{F}$, and

$$\lim_{\gamma \to 0} \text{dist}\,(\mathcal{F}, \mathcal{F}_\gamma) = 0. \tag{3.13}$$

Let $\boldsymbol{x}_{\text{pr}}^{\overline{k}(\gamma)}$ be the projection of $\boldsymbol{x}^{\overline{k}(\gamma)}$ onto $\mathcal{F}$. Then we have

$$
\begin{aligned}
0 &< V_P(T) - V_P(E^{\overline{k}(\gamma)}) \\
&= \left[\boldsymbol{c}^\top \boldsymbol{x}_{\text{pr}}^{\overline{k}(\gamma)} - V_P(E^{\overline{k}(\gamma)}) \right] - \left[\boldsymbol{c}^\top \boldsymbol{x}_{\text{pr}}^{\overline{k}(\gamma)} - V_P(T) \right] \\
&\leq \boldsymbol{c}^\top \boldsymbol{x}_{\text{pr}}^{\overline{k}(\gamma)} - \boldsymbol{c}^\top \boldsymbol{x}^{\overline{k}(\gamma)} \\
&\leq \|\boldsymbol{c}\| \, \|\boldsymbol{x}_{\text{pr}}^{\overline{k}(\gamma)} - \boldsymbol{x}^{\overline{k}(\gamma)}\|,
\end{aligned}
\tag{3.14}
$$

where the second inequality follows from $\boldsymbol{x}_{\text{pr}}^{\overline{k}(\gamma)} \in \mathcal{F}$. Since $\boldsymbol{x}^{\overline{k}(\gamma)} \in \mathcal{F}_\gamma$, we have

$$\|\boldsymbol{x}_{\text{pr}}^{\overline{k}(\gamma)} - \boldsymbol{x}^{\overline{k}(\gamma)}\| = \text{dist}(\boldsymbol{x}^{\overline{k}(\gamma)},\, \mathcal{F}) \leq \text{dist}\,(\mathcal{F}, \mathcal{F}_\gamma),$$

which together with (3.13) and (3.14) yields $\lim_{\gamma \to 0} V_P(E^{\overline{k}(\gamma)}) = V_P(T)$. The second statement of the theorem can be proved easily by using the boundedness of $\{\boldsymbol{x}^{\overline{k}(\gamma)}\}$ (Assumption B(a)). $\blacksquare$

Remark In Step 1 of Algorithm 3.1, we can choose p different points $\{t_1^k, \ldots, t_p^k\}$ such that $\boldsymbol{a}(t_i^k)^\top \boldsymbol{x}^k - b(t_i^k) < -\gamma$ for $i = 1, \ldots, p$, and let $\overline{E}^{k+1} := E^k \cup \{t_1^k, \ldots, t_p^k\}$. For such a multiple exchange algorithm, Theorems 3.1 and 3.2 can be proved in a similar manner.

4 Numerical Results

In this section, we report some numerical results. To implement Algorithm 3.1 practically, we set the details as follows. In Step 0, we set $\gamma := 10^{-6}$, and solve LSOCP(E^0) by the SOCCP algorithm proposed in [15], where each component of the initial point is chosen from $[-1, 1]$ randomly. In Step 1, we calculate t_{new}^k by combining the bisection method with Newton's method. In Step 2, we solve LSOCP($\overline{E}^{k+1}$) and DLSOCP($\overline{E}^{k+1}$) simultaneously by the same SOCCP algorithm, where the initial point is chosen by exploiting the solutions of the previous subproblems. For a linear SIP with $\Re_+^n$ or $\Re^n$ instead of $\mathcal{K}^n$, it is known that the simplex method is efficient for solving subproblems. However, as far as we know, there are no simplex type algorithms suitable for LSOCP($\overline{E}^k$) and DLSOCP($\overline{E}^k$). In Step 3, we relax the positivity criterion $\nu_{k+1}(t) > 0$ to $\nu_{k+1} > 10^{-6}$. The algorithms are coded in MATLAB 7.0 and run on a machine with Intel(R) Xeon(TM) CPU 3.60GHz and 2GB RAM.

In the first experiment, we solve the LSSIP (1.2) with $T = [-1, 1]$,

$$\boldsymbol{a}(t) := \begin{pmatrix} \alpha_{1,3} t^3 + \alpha_{1,2} t^2 + \alpha_{1,1} t - 1 \\ \alpha_{2,3} t^3 + \alpha_{2,2} t^2 + \alpha_{2,1} t \\ \vdots \\ \alpha_{n,3} t^3 + \alpha_{n,2} t^2 + \alpha_{n,1} t \end{pmatrix}, \text{ and } b(t) := -(\beta_1 t + \beta_2)^2 - (\beta_3 + 3), \tag{4.1}$$

where $\alpha_{i,j}$, β_j $(i = 1, \ldots, n,\ j = 1, 2, 3)$, and all components of $\boldsymbol{c}$ are randomly chosen from $[-2, 2]$. Then, for any choices of $\alpha_{i,j}$ and β_j, the feasible region of LSSIP (1.2) is compact and has nonempty interior. This can be observed as follows. Since $\boldsymbol{a}(0) = -(1, 0, \ldots, 0)^\top$ and $-b(0) > 0$, the feasible region of LSOCP($\{0\}$) is nonempty and compact. Moreover, for a sufficiently small $\varepsilon > 0$, the vector $\boldsymbol{u}_\varepsilon := (\varepsilon, 0, \ldots, 0)^\top$ satisfies $\boldsymbol{u}_\varepsilon \in \operatorname{int} \mathcal{K}^n$ and $\boldsymbol{a}(t)^T \boldsymbol{u}_\varepsilon - b(t) > 0$ for any $t \in T$. In this experiment, we generate 100 test problem instances for each $n = 100, 200, \ldots, 500$ and solve them by the proposed algorithm with $E^0 = \{0\}$. The obtained results are shown in Table 1, in which $\lambda_1(\boldsymbol{x}^*)$ and $\lambda_2(\boldsymbol{x}^*)$ denote the spectral values of the solution $\boldsymbol{x}^*$, $\sharp$ite denotes the number of iterations, and cpu(s) denotes the CPU time in seconds. All values are averages of 100 runs for each n. As the table shows, the algorithm finds the optimal solutions on the boundary of SOCs for all trials.[3] It is also seen that the computational cost becomes higher and higher as n becomes larger, whereas the number of iteration does not depend on the size of problems. Actually, we obtained the solution in 3 iterations for one problem and in 14 iterations for another problem, but the average was around 7 for every n.

In the second experiment, we solve the following two LSSIPs:

$$
\begin{aligned}
\text{Minimize} \quad & \sum_{i=1}^{7} \frac{x_i} {i} \\
\text{subject to} \quad & \boldsymbol{x} \in \mathcal{K}^7, \ \sum_{i=1}^{7} t^{i-1} x_i \geq \sum_{i=0}^{4} t^{2i} \quad (\forall\, t \in [0, 1]).
\end{aligned}
\tag{4.2}
$$

$$
\begin{aligned}
\text{Minimize} \quad & h \\
\text{subject to} \quad & \begin{pmatrix} h \\ \boldsymbol{x} \end{pmatrix} \in \mathcal{K}^8, \ h \geq \left| \sum_{i=1}^{7} t^{i-1} x_i - \sin\left(\frac{5\pi t}{6}\right) \right| \quad (\forall\, t \in [0, 1]).
\end{aligned}
\tag{4.3}
$$

Note that problem (4.3) is equivalent to the 7-dimensional unconstrained minimization problem with objective function $f(\boldsymbol{x}) := \max_{t \in [0,1]} \{ \|\boldsymbol{x}\|, \ |\sum_{i=1}^{7} t^{i-1} x_i - \sin(5\pi t/6)| \}$. Although (4.3) is not of the form LSSIP (1.2), it reduces to LSSIP (1.2) by setting $\boldsymbol{c} = (1, 0, \ldots, 0)^\top \in \Re^8$, $T = [0, 1] \cup [2, 3]$, and

$$
\boldsymbol{a}(t) := \begin{cases} \left(1,\, 1,\, t,\, t^2, \ldots,\, t^6\right)^\top & \text{if } t \in [0, 1] \\ \left(1,\, -1,\, -(t-2),\, -(t-2)^2, \ldots,\, -(t-2)^6\right)^\top & \text{if } t \in [2, 3] \\ \boldsymbol{0} & \text{otherwise}, \end{cases}
$$

$$
b(t) := \begin{cases} \sin\left(\dfrac{5\pi t}{6}\right) & \text{if } t \in [0, 1] \\ -\sin\left(\dfrac{5\pi(t-2)}{6}\right) & \text{if } t \in [2, 3] \\ -\infty & \text{otherwise}. \end{cases}
$$

The obtained results are shown in Table 2, where the values of $\sharp$ite and cpu(s) are the average of 100 runs with random choices of E^0 such that $|E^0| = n$ and initial points in solving LSOCP(E^0). We observe that, for all trials in LSSIP (4.2), the number of iteration is always 1. Moreover, at the solution $\boldsymbol{x}^*$, the active index set $E^* := \{ t \in T \mid \boldsymbol{a}(t)^\top \boldsymbol{x}^* - b(t) = 0 \}$ is the singleton $\{1\}$, which is the extreme point of T. On the other hand, for LSSIP (4.3), the number of iterations is between 2 and 4, and it varies with the choice of E^0 and initial points. Moreover, at the solution $\boldsymbol{x}^*$, we have $E^* = \{0.540\}$, which is not an extreme point of T.

---

[3]If vector $\boldsymbol{c}$ is chosen artificially, the solution can be at the origin or in the interior of SOC.

Table 1: Obtained results for LSSIP with (4.1)

n	$\lambda_1(\boldsymbol{x}^*)$	$\lambda_2(\boldsymbol{x}^*)$	$\sharp$ite	cpu(s)
100	0	8.12	7.31	2.81
200	0	8.61	6.85	5.88
300	0	8.49	7.14	13.17
400	0	8.07	7.33	23.90
500	0	8.71	7.55	39.53

Table 2: Obtained results for LSSIPs (4.2) and (4.3)

LSSIP	$\lambda_1(\boldsymbol{x}^*)$	$\lambda_2(\boldsymbol{x}^*)$	$\sharp$ite	cpu(s)
(4.2)	0	3.27	1	0.20
(4.3)	0	0.90	2.98	0.36

5   Concluding Remarks

We proposed the explicit exchange algorithm for solving linear SIP with SOC constraints, and proved the finite termination. Particularly, in the analyses, we adopted the special coordinate system based on the spectral factorization for SOCs. We also observed by some numerical experiments that the proposed algorithm finds the solution efficiently.

Our study is still in the infancy, and several issues remain to be addressed. In the paper, we considered only a single SOC for the constraint of $\boldsymbol{x} \in \mathcal{K}^n$. However, in the practical situation, we are often required to consider the multiple SOCs version, i.e., $\boldsymbol{x} \in \mathcal{K}^{n_1} \times \mathcal{K}^{n_2} \times \cdots \times \mathcal{K}^{n_m}$. Although we expect that the algorithm can be extended to multiple SOCs version in a natural fashion, the convergence analyses will be quite complicated. It is also challenging to develop the special algorithms for solving each SOCP. One may think of extending the simplex method for LP to SOCP. However, it is difficult in general since the feasible region of linear SOCP is not polyhedral and has an infinite number of extreme points.

References

[1] F. ALIZADEH AND D. GOLDFARB, *Second-order cone programming*, Mathematical Programming, 95 (2003), pp. 3–51.

[2] E. J. ANDERSON AND S.-Y. WU, *The continuous complementarity problem*, Optimization, 22 (1991), pp. 419–426.

[3] A. BEN-TAL AND A. NEMIROVSKI, *Robust convex optimization*, Mathematics of Operations Research, 23 (1998), pp. 769–805.

[4] ——, *Robust solutions of uncertain linear programs*, Operations Research Letters, 25 (1999), pp. 1–13.

[5] J.-S. CHEN, X. CHEN, AND P. TSENG, *Analysis of nonsmooth vector-valued functions associated with second-order cones*, Mathematical Programming, 101 (2004), pp. 95–117.

[6] I. D. COOPE AND C. J. PRICE, *Exact penalty function methods for nonlinear semi-infinite programming*, in Semi-Inifinte Programming, R. Reemtsen and J. Rückmann, eds., Kluwer Academic Publishers, Boston, 1998, pp. 137–157.

[7] R. W. COTTLE, J.-S. PANG, AND R. E. STONE, *The Linear Complementarity Problem*, Academic Press, San Diego, 1992.

[8] F. FACCHINEI AND J.-S. PANG, *Finite-Dimensional Variational Inequalities and Complementarity Problems*, Springer-Verlag, New York, 2003.

[9] J. FARAUT AND A. KORÁNYI, *Analysis on Symmetric Cones*, Clarendon Press, New York, 1994.

[10] M. FUKUSHIMA, Z.-Q. LUO, AND P. TSENG, *Smoothing functions for second-order cone complementarity problems*, SIAM Journal on Optimization, 12 (2001), pp. 436–460.

[11] M. A. GOBERNA AND M. A. LOPÉZ, *Semi-Infinite Programming: Recent Advances*, Kluwer Academic Publishers, Dordrecht, 2001.

[12] ——, *Linear semi-infinite programming: An updated survey*, European Journal of Operational Research, 143 (2002), pp. 390–405.

[13] G. GRAMLICH, R. HETTICH, AND E. SACHS, *Local convergence of SQP-methods in semi-infinite programming*, SIAM Journal on Optimization, 5 (1995), pp. 641–658.

[14] S. HAYASHI, T. YAMAGUCHI, N. YAMASHITA, AND M. FUKUSHIMA, *A matrix splitting method for symmetric affine second-order cone complementarity problems*, Journal of Computational and Applied Mathematics, 175 (2005), pp. 335–353.

[15] S. HAYASHI, N. YAMASHITA, AND M. FUKUSHIMA, *A combined smoothing and regularization method for monotone second-order cone complementarity problems*, SIAM Journal on Optimization, 15 (2005), pp. 593–615.

[16] R. HETTICH AND K. O. KORTANEK, *Semi-infinite programming: theory, methods, and applications*, SIAM Review, 35 (1993), pp. 380–429.

[17] H. C. LAI AND S.-Y. WU, *Extremal points and optimal solutions for general capacity problems*, Mathematical Programming, 54 (1992), pp. 87–113.

[18] ——, *On linear semi-infinite programming problems: an algorithm*, Numerical Functional Analysis and Optimization, 13 (1992), pp. 287–304.

[19] C. T. LAWRENCE AND A. L. TITS, *Feasible sequential quadratic programming for finely discretized problems from SIP*, in Semi-Inifinte Programming, R. Reemtsen and J. Rückmann, eds., Kluwer Academic Publishers, Boston, 1998, pp. 159–193.

[20] M. S. LOBO, L. VANDENBERGHE, S. BOYD, AND H. LEBRET, *Applications of second-order cone programming*, Linear Algebra and Its Applications, 284 (1998), pp. 193–228.

[21] M. LOPÉZ AND G. STILL, *Semi-infinite programming*, European Journal of Operational Research, to appear.

[22] M. MURAMATSU, *A pivoting procedure for a class of second-order cone programming*, Optimization Methods and Software, 21 (2006), pp. 295–314.

[23] E. POLAK, *Optimization: Algorithms and Consistent Approximation*, Springer, New York, 1997.

[24] R. REEMSTEN AND J. RÜCKMANN, *Semi-Infinite Programming*, Kluwer Academic Publishers, Boston, 1998.

[25] Y. TANAKA, M. FUKUSHIMA, AND T. IBARAKI, *A globally convergent SQP method for semi-infinite nonlinear optimization*, Journal of Computational and Applied Mathematics, 23 (1988), pp. 141–153.

[26] K. L. TEO, X. Q. YANG, AND L. S. JENNINGS, *Computational discretization algorithms for functional inequality constrained optimization*, Annals of Operations Research, 98 (2000), pp. 215–234.

[27] K. C. TOH, R. H. TÜTÜNCÜ, AND M. J. TODD, *SDPT3 version 3.02 – a MATLAB software for semidefinite-quadratic-linear programming*, updated in December 2002. `http://www.math.nus.edu.sg/~mattohkc/sdpt3.html`.

[28] S.-Y. WU, S. C. FANG, AND C. J. LIN, *Relaxed cutting plane method for solving semi-infinite programming problems*, Journal of Optimization Theory and Applications, 99 (1998), pp. 759–779.

[29] S.-Y. WU, D. H. LI, L. QI, AND G. ZHOU, *An iterative method for solving KKT system of the semi-infinite programming*, Optimization Methods and Software, 20 (2005), pp. 629–643.

[30] H. YAMASHITA AND H. YABE, *A primal-dual interior point method for nonlinear optimization over second order cones*, report, 2005. `http://www.optimization-online.org/DB_HTML/2005/07/1170.html`.