Local reduction based SQP-type method for semi-infinite programs with an infinite number of second-order cone constraints*

Takayuki Okuno and Masao Fukushima†

October 29, 2012

Abstract

The second-order cone program (SOCP) is an optimization problem with second-order cone (SOC) constraints and has achieved notable developments in the last decade. The classical semi-infinite program (SIP) is represented with infinitely many inequality constraints, and has been studied extensively so far. In this paper, we consider the SIP with infinitely many SOC constraints, called the SISOCP for short. Compared with the standard SIP and SOCP, the studies on the SISOCP are scarce, even though it has important applications such as Chebychev approximation for vector-valued functions. For solving the SISOCP, we develop an algorithm that combines a local reduction method with an SQP-type method. In this method, we reduce the SISOCP to an SOCP with finitely many SOC constraints by means of implicit functions and apply an SQP-type method to the latter problem. We study the global and local convergence properties of the proposed algorithm. Finally, we observe the effectiveness of the algorithm through some numerical experiments.

Keywords. semi-infinite programming; second-order cone constraints; SQP-type method; local reduction method

1 Introduction

In this paper, we focus on the following semi-infinite program with an infinite number of second-order cone constraints (SISOCP):

$$\begin{aligned} &\underset{x\in\mathbb{R}^n}{\text{Minimize}} && f(x) \\ &\text{subject to} && g(x,t)\in\mathcal{K}^m \quad \text{for all } t\in T, \end{aligned} \tag{1.1}$$

where $f:\mathbb{R}^n\to\mathbb{R}$ and $g:\mathbb{R}^n\times\mathbb{R}^\ell\to\mathbb{R}^m$ are twice continuously differentiable functions, and T is a nonempty *compact* index set given by

$$T := \{t\in\mathbb{R}^\ell \mid h_i(t)\geq 0,\ i=1,2,\dots,p\},$$

*This research was supported in part by Grant-in-Aid for Scientific Research from Japan Society for the Promotion of Science.

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

where $h_i : \mathbb{R}^\ell \to \mathbb{R}$ are twice continuously differentiable functions for $i = 1, 2, \ldots, p$. Moreover, $\mathcal{K}^m \subseteq \mathbb{R}^m$ denotes the m-dimensional second-order cone (SOC) defined by

$$\mathcal{K}^m := \begin{cases} \{(z_1, \tilde{z}^\top)^\top \in \mathbb{R} \times \mathbb{R}^{m-1} \mid z_1 \geq \|\tilde{z}\|\} & (m \geq 2) \\ \mathbb{R}_+ := \{z \in \mathbb{R} \mid z \geq 0\} & (m = 1). \end{cases}$$

We consider the problem (1.1) that contains a single SOC with $m \geq 2$ for simplicity of expression, although we can deal with the more general SISOCP that contains multiple SOCs as well as equality constraints, i.e.,

$$\begin{array}{ll} \underset{x \in \mathbb{R}^n}{\text{Minimize}} & f(x) \\ \text{subject to} & g^0(x) = 0, \\ & g^s(x,t) \in \mathcal{K}^{m_s} \quad \text{for all } t \in T^s \ (s = 1, 2, \ldots, S), \end{array} \tag{1.2}$$

where $g^0 : \mathbb{R}^n \to \mathbb{R}^{m_0}$ and $g^s : \mathbb{R}^n \times \mathbb{R}^{\ell_s} \to \mathbb{R}^{m_s}$ $(s = 1, 2, \ldots, S)$ are twice continuously differentiable functions, and $T^s \subseteq \mathbb{R}^{\ell_s}$ $(s = 1, 2, \ldots, S)$ are nonempty compact index sets given by $T^s := \{t \in \mathbb{R}^{\ell_s} \mid h_i^s(t) \geq 0, \ i = 1, 2, \ldots, p_s\}$ with twice continuously differentiable functions $h_i^s : \mathbb{R}^{\ell_s} \to \mathbb{R}$ $(i = 1, 2, \ldots, p_s)$. It is possible to extend the subsequent analysis for (1.1) to the general SISOCP (1.2) in a direct manner. In fact, we will show some numerical results for SISOCPs that contain multiple SOCs; see Experiment 3 in Section 5.

When T is a finite set, SISOCP (1.1) is nothing but a standard second-order cone program (SOCP) [14, 1]. Studies on the SOCP have been advanced significantly in the last decade. Especially, development of research on the linear SOCP (LSOCP) is notable. The primal-dual interior-point method [14, 1] is well known as an effective algorithm for solving LSOCP, and some software packages implementing them [22, 24] have been produced. The nonlinear SOCP (NLSOCP) is more complicated and has been studied not so much as LSOCP [11, 25, 27]. The second-order cone complementarity problem (SOCCP) is another important problem involving SOCs. The Karush-Kuhn-Tucker conditions for LSOCP and NLSOCP are particularly represented as SOCCPs. The smoothing method [8, 4] is one of useful algorithms for solving the SOCCPs.

When $m = 1$ and $\mathcal{K} = \mathbb{R}_+$, SISOCP (1.1) reduces to the classical semi-infinite program (SIP) [5, 10, 12, 15, 21]. The SIP has received much attention of many researchers so far. It has wide applications in engineering, e.g., the air pollution control, the robot trajectory planning, the stress of materials, and so on [10, 15].

One of important applications of SISOCP (1.1) is a vector-valued Chebyshev approximation problem. Let $X \subseteq \mathbb{R}^\ell$ be a nonempty set, $Y \subseteq \mathbb{R}^n$ be a given compact set, and $\Phi : Y \to \mathbb{R}^m$ and $F : \mathbb{R}^\ell \times Y \to \mathbb{R}^m$ be given functions. Then, the Chebyshev approximation problem is to find a parameter $u \in X$ such that $\Phi(y) \approx F(u, y)$ for all $y \in Y$. One relevant approach is to solve the following problem:

$$\underset{u \in X}{\text{Minimize}} \quad \max_{y \in Y} \|\Phi(y) - F(u, y)\|. \tag{1.3}$$

By introducing the auxiliary variable $r \in \mathbb{R}$, we can reformulate the above problem as

$$\begin{array}{ll} \underset{(u,r) \in X \times \mathbb{R}}{\text{Minimize}} & r \\ \text{subject to} & \begin{pmatrix} r \\ \Phi(y) - F(u, y) \end{pmatrix} \in \mathcal{K}^m \quad \text{for all } y \in Y, \end{array}$$

which is of the form (1.1). See also [16] for another application of SISOCP (1.1).

For solving the standard SIP, there exist many algorithms such as the discretization method [9, 20], the exchange method [7, 12, 26, 16], the local reduction method [6, 23, 17, 18], and others

[3, 13, 19, 26]. The discretization method solves a sequence of relaxed SIPs with T replaced by $T^k \subseteq T$, where T^k is a finite index set such that the distance[1] from T^k to T tends to 0 as k goes to infinity. While this method is comprehensible and easy to implement, the computational cost tends to be high since the cardinality of T^k grows exponentially in the dimension of T. The exchange method solves a relaxed subproblem with T replaced by a finite subset $T^k \subseteq T$, where T^k is updated so that $T^{k+1} \subseteq T^k \cup \{t_1, t_2, \cdots, t_r\}$ with some $\{t_1, t_2, \cdots, t_r\} \subseteq T \setminus T^k$. In the local reduction based method, an infinite number of constraints in the original SIP are rewritten as a finite number of constraints by using implicit functions. Then, the obtained standard nonlinear program is solved by existing methods.

In contrast with the SIP, numerical methods for SISOCP (1.1) have not been studied so much. Recently, Okuno, Hayashi and Fukushima [16] proposed a regularized explicit exchange method to solve semi-infinite programs with infinitely many conic constraints. Although an exchange-type algorithm is effective to find an approximate solution, it is not very suitable to obtain an accurate solution. On the other hand, a local reduction-type method is known to have an advantage in computing an accurate solution with fast convergence speed [6, 23, 17, 18]. In this paper, for solving SISOCP (1.1), we propose a local reduction based method combined with a sequential quadratic programming (SQP) method, where, in each iteration, we replace the SISOCP with an NLSOCP by means of the local reduction method and then generate a search direction by solving a quadratic SOCP that approximates the NLSOCP.

This paper is organized as follows. In Section 2, we study the local reduction method for SISOCP (1.1). We define some concepts and give important propositions to represent the SISOCP locally as an NLSOCP by using implicit functions. In Section 3, we propose an SQP-type method combined with the local reduction method for solving SISOCP (1.1). In Section 4, we analyze the global and local convergence properties of the proposed algorithm. In Section 5, we observe the effectiveness of the algorithm by some numerical experiments. In Section 6, we conclude the paper with some remarks.

Throughout the paper, we use the following notations. The 2-norm of a vector $z \in \mathbb{R}^m$ are defined by $\|z\| := \sqrt{z^\top z}$. The symbol $\perp$ means the perpendicularity. For any vector $z \in \mathbb{R}^m$, we let $(z)_+ := \max(z, 0)$, where the maximum is taken componentwise. For $z^i \in \mathbb{R}^{m_i}$ $(i = 1, 2, \ldots, p)$, we often write $(z^1, z^2, \cdots, z^p)$ to denote $((z^1)^\top, (z^2)^\top, \cdots, (z^p)^\top)^\top \in \mathbb{R}^{m_1} \times \mathbb{R}^{m_2} \times \cdots \times \mathbb{R}^{m_p}$. For any vector $z \in \mathbb{R}^n$ and any vector function $G : \mathbb{R}^n \to \mathbb{R}^m$, we let $z = (z_1, \tilde{z}) \in \mathbb{R} \times \mathbb{R}^{n-1}$ and $G(z) = (G_1(z), \tilde{G}(z)) \in \mathbb{R} \times \mathbb{R}^{m-1}$. For any twice continuously differentiable vector function $H : \mathbb{R}^n \to \mathbb{R}^m$ and any vector $w \in \mathbb{R}^m$, we denote $\nabla^2 H(z) w := \sum_{i=1}^m w_i \nabla^2 H_i(z)$, where $H(z) := (H_1(z), H_2(z), \ldots, H_m(z))^\top \in \mathbb{R}^m$. For any scalar function $\psi : \mathbb{R}^n \to \mathbb{R}$ and $z \in \mathbb{R}^n$, we let $\psi_+(z) := (\psi(z))_+$. We denote the n-dimensional identity matrix and the set of $n \times n$ symmetric positive definite matrices by I_n and S^n_{++}, respectively. For $z \in \mathbb{R}^n$, $\mathrm{diag}(z)$ denotes the $n \times n$ diagonal matrix with diagonal elements z_i $(i = 1, 2, \ldots, n)$.

2 Local reduction of SISOCP to SOCP

In this section, we study the local reduction method for SISOCP (1.1). In relation to the constraints in SISOCP (1.1), we first consider the problem

$$P(x): \quad \begin{array}{ll} \underset{t \in \mathbb{R}^\ell}{\text{Minimize}} & \lambda(x, t) := g_1(x, t) - \|\tilde{g}(x, t)\| \\ \text{subject to} & t \in T = \{t \in \mathbb{R}^\ell \mid h_i(t) \geq 0,\ i = 1, 2, \ldots, p\}, \end{array} \tag{2.1}$$

[1]For two sets $X \subseteq Y$, the distance from X to Y is defined as $\mathrm{dist}\,(X, Y) := \sup_{y \in Y} \inf_{x \in X} \|x - y\|$.

where $g_1(x,t)$ is the first component of $g(x,t) \in \mathbb{R}^m$ and $\tilde{g}(x,t)$ is the vector consisting of the remaining $m-1$ components of $g(x,t)$. We call problem (2.1) the lower-level problem of SISOCP (1.1) and let

$$\varphi(x) := \max_{t \in T} \left(-\lambda(x,t)\right). \tag{2.2}$$

Obviously, the infinitely many SOC constraints $g(x,t) \in \mathcal{K}^m$ $(t \in T)$ are equivalent to the condition $\varphi(x) \leq 0$. Hence, SISOCP (1.1) can be rewritten equivalently as

$$\underset{x \in \mathbb{R}^n}{\text{Minimize}} \;\; f(x) \;\text{ subject to } \varphi(x) \leq 0. \tag{2.3}$$

Though problem (2.3) has only one constraint, treating $\varphi(x) \leq 0$ directly is difficult since it is not differentiable everywhere. As a remedy, we take the local reduction method. In this method, at any $\bar{x} \in \mathbb{R}^n$, we find an open neighborhood $U(\bar{x}) \subseteq \mathbb{R}^n$ of $\bar{x}$ and continuously differentiable functions $t_j : U(\bar{x}) \to T$ $(j = 1, 2, \ldots, r(\bar{x}))$ such that

$$\varphi(x) = \max_{1 \leq j \leq r(\bar{x})} \left(-\lambda(x, t_j(x))\right)$$

holds for all $x \in U(\bar{x})$, where each $t_j(x)$ represents a local maximum of $-\lambda(x,t)$ on T and $r(\bar{x})$ is a positive integer. This means that $\varphi(x) \geq 0$ may be reduced to the finitely many SOC constraints $g(x, t_j(x)) \in \mathcal{K}^m$ $(j = 1, 2, \ldots, r)$ in the set $U(\bar{x})$, i.e., problem (2.3) can be transformed locally to

$$\begin{aligned} &\underset{x \in U(\bar{x})}{\text{Minimize}} && f(x) \\ &\text{subject to} && g(x, t_j(x)) \in \mathcal{K}^m \quad (j = 1, 2, \ldots, r(\bar{x})). \end{aligned} \tag{2.4}$$

Then, we can expect that existing methods such as an SQP-type method [11] work efficiently for solving the reduced SOCP (2.4).

To give more formal treatment of the local reduction method, let $l : \mathbb{R}^n \times \mathbb{R}^\ell \times \mathbb{R}^p \to \mathbb{R}$ denote the Lagrangian of the lower-level problem $P(x)$, i.e.,

$$l(x,t,\alpha) := \lambda(x,t) - h(t)^\top \alpha,$$

where $h(t) := (h_1(t), h_2(t), \ldots, h_p(t))^\top$, and $\alpha := (\alpha_1, \alpha_2, \ldots, \alpha_p)^\top \in \mathbb{R}^p$ is a Lagrange multiplier vector corresponding to the constraints $h_i(t) \geq 0$ $(i = 1, 2, \ldots, p)$. Let $I_a(t)$ denote the active index set at $t \in \mathbb{R}^\ell$, i.e.,

$$I_a(t) := \{i \in \{1, 2, \ldots, p\} \mid h_i(t) = 0\}. \tag{2.5}$$

We define the *nondegeneracy* of local optima of $P(x)$.

Definition 2.1 (Nondegeneracy)**.** *Let $\bar{x} \in \mathbb{R}^n$, and let $\bar{t} \in T$ be a local optimum of $P(\bar{x})$ such that the linear independence constraint qualification holds, i.e., $\{\nabla h_i(\bar{t})\}_{i \in I_a(\bar{t})}$ are linearly independent. Furthermore, suppose that the function $\lambda(\cdot,\cdot)$ is twice continuously differentiable at $(\bar{x}, \bar{t})$. Then, there exists a Lagrange multiplier vector $\alpha := (\alpha_1, \alpha_2, \ldots, \alpha_p)^\top \in \mathbb{R}^p$ such that*

$$\nabla_t l(\bar{x}, \bar{t}, \alpha) = 0, \quad 0 \leq \alpha \perp h(\bar{t}) \geq 0.$$

We say that $\bar{t} \in T$ is nondegenerate if

(a) *the second-order sufficient condition*

$$v^\top \nabla^2_{tt} l(\bar{x}, \bar{t}, \alpha) v > 0 \;\textit{ for all } v \in C(\bar{t}) \setminus \{0\}$$

with

$$C(t) := \begin{cases} \{v \in \mathbb{R}^\ell \mid v^\top \nabla h_i(t) = 0,\; i \in I_a(t)\} & (I_a(t) \neq \emptyset), \\ \mathbb{R}^\ell & (I_a(t) = \emptyset) \end{cases}$$

holds, and

(b) *the strict complementarity*

$$\alpha_i > 0 \ \text{for all } i \in I_a(\bar{t})$$

holds.

Under the nondegeneracy assumption, we have the following proposition.

Proposition 2.2. *Let $x \in \mathbb{R}^n$ and $\bar{x} \in \mathbb{R}^n$. Assume that $\bar{t} \in T$ is a nondegenerate local optimum of $P(\bar{x})$ and $\bar{\alpha} := (\bar{\alpha}_1, \bar{\alpha}_2, \ldots, \bar{\alpha}_p)^\top \in \mathbb{R}^p_+$ is a Lagrange multiplier vector corresponding to the constraints $h_i(t) \geq 0$ $(i = 1, 2, \ldots, p)$. Furthermore, suppose that $\lambda(\cdot, \cdot)$ is twice continuously differentiable at $(\bar{x}, \bar{t})$. Then, there exist an open neighborhood $U(\bar{x})$ of $\bar{x}$ and twice continuously differentiable functions $t(\cdot) : U(\bar{x}) \to T$ and $\alpha_i(\cdot) : U(\bar{x}) \to \mathbb{R}_+$ $(i = 1, 2, \ldots, p)$ such that*

(a) $t(\bar{x}) = \bar{t}$, $\alpha_i(\bar{x}) = \bar{\alpha}_i$ $(i \in I_a(\bar{t}))$ *and* $\alpha_i(\bar{x}) = 0$ $(i \notin I_a(\bar{t}))$,

(b) $t(x)$ *is a nondegenerate local optimum of* $P(x)$ *for each* $x \in U(\bar{x})$ *with a unique Lagrange multiplier vector* $(\alpha_1(x), \alpha_2(x), \ldots, \alpha_p(x))^\top \in \mathbb{R}^p_+$,

(c) $\nabla t(\bar{x}) \in \mathbb{R}^{n \times \ell}$ *and* $\nabla \alpha_i(\bar{x}) \in \mathbb{R}^n$ $(i \in I_a(\bar{t}))$ *comprise a unique solution of the linear system*

$$\begin{pmatrix} \nabla^2_{tt} l(\bar{x}, \bar{t}, \bar{\alpha}) & \nabla h_a(\bar{t}) \\ \nabla h_a(\bar{t})^\top & 0 \end{pmatrix} \begin{pmatrix} \nabla t(\bar{x})^\top \\ \nabla \alpha_a(\bar{x})^\top \end{pmatrix} = - \begin{pmatrix} \nabla^2_{tx} \lambda(\bar{x}, \bar{t}) \\ 0 \end{pmatrix}, \tag{2.6}$$

where

$$\nabla \alpha_a(\bar{x}) := (\nabla \alpha_i(\bar{x}))_{i \in I_a(\bar{t})} \in \mathbb{R}^{n \times |I_a(\bar{t})|}, \ \nabla h_a(\bar{t}) := (\nabla h_i(\bar{t}))_{i \in I_a(\bar{t})} \in \mathbb{R}^{\ell \times |I_a(\bar{t})|};$$

in particular, if $I_a(\bar{t}) = \emptyset$, then $\nabla t(\bar{x}) \in \mathbb{R}^{n \times \ell}$ is a unique solution of the linear system

$$\nabla^2_{tt} l(\bar{x}, \bar{t}, \bar{\alpha}) \nabla t(\bar{x})^\top = -\nabla^2_{tx} \lambda(\bar{x}, \bar{t}), \tag{2.7}$$

(d) *for any $x \in U(\bar{x})$, letting $v(x) := \lambda(x, t(x))$, we have*

$$\begin{aligned} \nabla v(\bar{x}) &= \nabla_x \lambda(\bar{x}, \bar{t}), \\ \nabla^2 v(\bar{x}) &= \nabla^2_{xx} \lambda(\bar{x}, \bar{t}) - \nabla t(\bar{x}) \nabla^2_{tt} l(\bar{x}, \bar{t}, \bar{\alpha}) \nabla t(\bar{x})^\top. \end{aligned}$$

Proof. Apply the implicit function theorem to the following equations:

$$\nabla_t l(x, t, \alpha) = 0, \ h_i(t) = 0 \ \ (i \in I_a(\bar{t})),$$

which come from the Karush-Kuhn-Tucker (KKT) conditions of $P(x)$. See also [10, 6]. □

Next, let

$$T_{\text{loc}}(x) := \{t \in T \mid t \ \text{ is a local optimum of } P(x)\}$$

and

$$T_\varepsilon(x) := T_{\text{loc}}(x) \cap \{t \in T \mid \lambda(x, t) \leq \min_{t \in T} \lambda(x, t) + \varepsilon\}$$

for a given constant $\varepsilon > 0$. Now, we show that the infinitely many SOC constraints $g(x, t) \in \mathcal{K}^m$ $(t \in T)$ can locally be represented as finitely many SOC constraints under some assumptions including the following *regularity* condition.

Definition 2.3 (Regularity)**.** *Let $x \in \mathbb{R}^n$ and $\varepsilon > 0$ be given. We say that x is regular if any $t \in T_\varepsilon(x)$ is nondegenerate and $|T_\varepsilon(x)| < \infty$.*

Proposition 2.4. *Let $\bar{x} \in \mathbb{R}^n$ and $\varepsilon > 0$ be given. Suppose that $\bar{x}$ is regular and let $T_\varepsilon(\bar{x}) := \{\bar{t}_1, \bar{t}_2, \ldots, \bar{t}_{r_\varepsilon(\bar{x})}\}$. Furthermore, suppose that the function $\lambda(\cdot,\cdot)$ is twice continuously differentiable at $(\bar{x}, \bar{t}_j)$ for all $j = 1, 2, \ldots, r_\varepsilon(\bar{x})$ and that $\bar{x} \in \mathbb{R}^n$ is regular. Then, there exist an open neighborhood $U_\varepsilon(\bar{x}) \subseteq \mathbb{R}^n$ of $\bar{x}$ and functions $t_1(\cdot), t_2(\cdot), \cdots, t_{r_\varepsilon(\bar{x})}(\cdot) : U_\varepsilon(\bar{x}) \to T$ such that, for each $j = 1, 2, \ldots, r_\varepsilon(\bar{x})$,*

(a) *$t_j(\cdot)$ is twice continuously differentiable,*

(b) *$t_j(\bar{x}) = \bar{t}_j$, and*

(c) *$\varphi(x) = \max_{j=1,2,\cdots,r_\varepsilon(\bar{x})}(-\lambda(x, t_j(x)))$ for all $x \in U_\varepsilon(\bar{x})$, where $\varphi(\cdot)$ is defined by (2.2).*

Moreover, SISOCP (1.1) can locally be reduced to the following SOCP:

$$\begin{aligned} &\underset{x \in U_\varepsilon(\bar{x})}{\text{Minimize}} && f(x) \\ &\text{subject to} && g(x, t_j(x)) \in \mathcal{K}^m, \ (j = 1, 2, \ldots, r_\varepsilon(\bar{x})). \end{aligned}$$

Proof. We omit the proof since it can easily be derived from Proposition 2.2 and the implicit function theorem. □

3 Local reduction based SQP-type algorithm for the SISOCP

The Karush-Kuhn-Tucker (KKT) conditions for SISOCP (1.1) are represented as follows [16] : Let x^* be a local optimum of SISOCP (1.1). Then, under suitable constraint qualification, there exist q indices $t_1^*, t_2^*, \ldots, t_q^* \in T_\varepsilon(x^*)$ and Lagrange multipliers $\eta_1^*, \eta_2^*, \ldots, \eta_q^* \in \mathbb{R}^m$ such that $q \le n$ and

$$\nabla f(x^*) - \sum_{j=1}^{q} \nabla_x g(x^*, t_j^*)\eta_j^* = 0, \tag{3.1}$$

$$\mathcal{K}^m \ni \eta_j^* \perp g(x^*, t_j^*) \in \mathcal{K}^m \ (j = 1, 2, \ldots, q). \tag{3.2}$$

In this section, we propose an algorithm for finding a vector $x^* \in \mathbb{R}^n$ that satisfies the above KKT conditions. In the algorithm, we combine the local reduction method with the sequential quadratic programming (SQP) method. Let $\varepsilon > 0$ be given and let $x^k \in \mathbb{R}^n$ be a current iterate. Assume that x^k satisfies the *regularity* defined in Definition 2.3. Then, from Proposition 2.4, there exist some open neighborhood $U_\varepsilon(x^k) \subseteq \mathbb{R}^n$ of x^k and twice continuously differentiable functions $t_j^k : U_\varepsilon(x^k) \to T$ $(j = 1, 2, \ldots, r_\varepsilon(x^k))$ such that SISOCP (1.1) can locally be reduced to the following SOCP:

$$\text{SOCP}(x^k, \varepsilon) : \quad \begin{aligned} &\underset{x \in U_\varepsilon(x^k)}{\text{Minimize}} && f(x) \\ &\text{subject to} && G_j^k(x) := g(x, t_j^k(x)) \in \mathcal{K}^m \ (j = 1, 2, \ldots, r_\varepsilon(x^k)). \end{aligned}$$

We then generate a search direction $d^k \in \mathbb{R}^n$ by solving the following Quadratic SOCP (QSOCP), which consists of quadratic and linear approximations of the objective function and constraint functions of SOCP(x^k, ε), respectively:

$$\text{QSOCP}(x^k, \varepsilon) : \quad \begin{aligned} &\underset{d \in \mathbb{R}^n}{\text{Minimize}} && \nabla f(x^k)^\top d + \tfrac{1}{2} d^\top B_k d \\ &\text{subject to} && G_j^k(x^k) + \nabla G_j^k(x^k)^\top d \in \mathcal{K}^m \ (j = 1, 2, \ldots, r_\varepsilon(x^k)), \end{aligned}$$

where $B_k \in \mathbb{R}^{n\times n}$ is a symmetric positive definite matrix. Note that $G_j^k(x^k)$ and $\nabla G_j^k(x^k)$ are given by

$$G_j^k(x^k) = g(x^k, t_j^k(x^k)), \tag{3.3}$$
$$\nabla G_j^k(x^k) = \nabla_x g(x^k, t_j^k(x^k)) + \nabla t_j^k(x^k)\nabla_t g(x, t_j^k(x^k)), \tag{3.4}$$

where $t_j^k(x^k) \in T_\varepsilon(x^k)$ and $\nabla t_j^k(x^k)$ can be obtained by solving the lower-level problem $P(x^k)$ and by solving (2.6) or (2.7). Under some constraint qualification, the optimum d^k of QSOCP(x^k,ε) satisfies the following KKT conditions:

$$\nabla f(x^k) + B_k d^k - \sum_{j=1}^{r_\varepsilon(x^k)} \nabla G_j^k(x^k)\eta_j^{k+1} = 0, \tag{3.5}$$
$$\mathcal{K}^m \ni \eta_j^{k+1} \perp G_j^k(x^k) + \nabla G_j^k(x^k)^\top d^k \in \mathcal{K}^m \ (j = 1,2,\ldots,r_\varepsilon(x^k)), \tag{3.6}$$

where $\eta_j^{k+1} \in \mathbb{R}^m$ $(j = 1,2,\ldots,r_\varepsilon(x^k))$ are Lagrange multiplier vectors corresponding to the SOC constraints $G_j^k(x^k)+\nabla G_j^k(x^k)^\top d \in \mathcal{K}^m$ $(j = 1,2,\ldots,r_\varepsilon(x^k))$. If $d^k = 0$, then it follows immediately from (3.5) and (3.6) that the KKT conditions for solving SOCP(x^k,ε) are satisfied at x^k. If, in addition, $\tilde{g}(x^k,t) \neq 0$ holds for all $t \in T_\varepsilon(x^k)$, then it holds that

$$\nabla f(x^k) - \sum_{j=1}^{r_\varepsilon(x^k)} \nabla_x g(x^k, \bar{t}_j^k)\eta_j^{k+1} = 0,$$
$$\mathcal{K}^m \ni \eta_j^{k+1} \perp g(x^k, \bar{t}_j^k) \in \mathcal{K}^m \ (j = 1,2,\ldots,r_\varepsilon(x^k)),$$

where $\bar{t}_j^k := t_j^k(x^k)$ $(j = 1,2,\ldots,r_\varepsilon(x^k))$, as will be shown by Proposition 4.2. These are actually regarded as the KKT conditions (3.1) and (3.2) of SISOCP (1.1). In particular, we can see that x^k is feasible for SISOCP (1.1), since $g(x^k,\bar{t}_j^k) \in \mathcal{K}^m$ for all $j = 1,2,\ldots,r_\varepsilon(x^k)$ and $\bar{t}_j^k$ $(j = 1,2,\ldots,r_\varepsilon(x^k))$ contain all minimizers of the lower-level problem $P(x^k)$.

To generate the next iterate x^{k+1} along the direction d^k, we need to choose a step size. To this end, we perform a line search with the following ℓ_∞-type penalty function:

$$\Phi_\rho(x) := f(x) + \rho\varphi_+(x), \tag{3.7}$$

where $\varphi(\cdot)$ is defined by (2.2) and $\rho > 0$ is a penalty parameter. Notice that the function $\Phi_\rho(\cdot)$ is continuous everywhere. Another plausible choice of a merit function used in the line search is an ℓ_1-type penalty function, i.e., the function (3.7) with $\varphi_+(x)$ $\left(= \max_{t\in T}\left(-\lambda(x,t)\right)_+\right)$ replaced by $\sum_{t\in T_\varepsilon(x)}\left(-\lambda(x,t)\right)_+$. However, in the semi-infinite case, the ℓ_1-type penalty function has such a serious drawback that it may fail to be continuous at a point where the cardinality of $T_\varepsilon(x)$ changes. Properties of penalty functions for semi-infinite programs are explained in detail together with a specific example in [23]. Now, we formally state the SQP-type algorithm for solving SISOCP (1.1).

Algorithm 1

Step 0 (Initialization): Choose $x^0 \in \mathbb{R}^n$ and a matrix $B_0 \in S_{++}^n$. Select parameters $\alpha \in (0,1)$, $\beta \in (0,1)$, $\delta > 0$, $\varepsilon > 0$ and $\rho_{-1} > 0$. Set $k := 0$.

Step 1 (Generate a search direction): Solve QSOCP(x^k,ε) to obtain $d^k \in \mathbb{R}^n$ and corresponding Lagrange multipliers $\eta_j^{k+1} \in \mathcal{K}^m$ $(j = 1,2,\ldots,r_\varepsilon(x^k))$.

Step 2 (Check convergence): If $d^k = 0$, then stop. Otherwise, go to Step 3.

Step 3 (Update penalty parameter): If $\rho_{k-1} \geq \sum_{j=1}^{r_\varepsilon(x^k)}(\eta_j^{k+1})_1$, then set $\rho_k := \rho_{k-1}$. Otherwise, set $\rho_k := \sum_{j=1}^{r_\varepsilon(x^k)}(\eta_j^{k+1})_1 + \delta$.

Step 4 (Armijo line search): Find the smallest nonnegative integer $r_k \geq 0$ satisfying

$$\Phi_{\rho_k}(x^k + \alpha^{r_k} d^k) - \Phi_{\rho_k}(x^k) \leq -\alpha^{r_k}\beta(d^k)^\top B_k d^k.$$

Set $s_k := \alpha^{r_k}$ and $x^{k+1} := x^k + s_k d^k$.

Step 5: Update the matrix B_k to obtain $B_{k+1} \in S^n_{++}$. Set $k := k+1$ and return to Step 1.

To construct QSOCP(x^k, ε) at each iteration k, we need to obtain the set $T_\varepsilon(x^k)$ by computing all local minimizers of the lower-level problem $P(x^k)$. Moreover, we have to compute $\nabla t^k_j(x^k)$ $(j = 1, 2, \ldots, r_\varepsilon(x^k))$ by solving the linear system (2.6) or (2.7). In Step 5, we must compute $\max_{t\in T}(-\lambda(x^k + \alpha^r d^k, t))_+$ to evaluate $\Phi_{\rho_k}(x^k + \alpha^r d^k)$ for each r. In Step 6, we may choose B_k as

$$B_k := \nabla^2 f(x^k) - \sum_{j=1}^{r_\varepsilon(x^k)} (\zeta^k_j)_1 W_{kj}, \tag{3.8}$$

where

$$W_{kj} := \nabla^2 G^k_{j1}(x^k) - \frac{\nabla^2 \tilde{G}^k_j(x^k)\tilde{G}^k_j(x^k)}{\|\tilde{G}^k_j(x^k)\|} \quad (j = 1, 2, \ldots, r_\varepsilon(x^k))$$

and ζ^k_j $(j = 1, 2, \ldots, r_\varepsilon(x^k))$ are some estimates of Lagrange multiplier vectors corresponding to the constraints $G^k_j(\cdot) \in \mathcal{K}^m$ $(j = 1, 2, \ldots, r_\varepsilon(x^k))$. A specific choice of ζ^k_j $(j = 1, 2, \ldots, r_\varepsilon(x^k))$ will be provided later in the section of numerical experiments. The matrix W_{kj} can be calculated as follows: Let $v^k_j : U_\varepsilon(x^k) \to \mathbb{R}$ be defined by $v^k_j(x) := G^k_{j1}(x) - \|\tilde{G}^k_j(x)\|$ for $j = 1, 2, \ldots, r_\varepsilon(x^k)$. Then, we have

$$\nabla^2 v^k_j(x^k) = W_{kj} - \frac{\nabla\tilde{G}^k_j(x^k)\nabla\tilde{G}^k_j(x^k)^\top}{\|\tilde{G}^k_j(x^k)\|} + \frac{\nabla\tilde{G}^k_j(x^k)\tilde{G}^k_j(x^k)\tilde{G}^k_j(x^k)^\top\nabla\tilde{G}^k_j(x^k)^\top}{\|\tilde{G}^k_j(x^k)\|^3},$$

which implies that, for $j = 1, 2, \ldots, r_\varepsilon(x^k)$,

$$W_{kj} = \nabla^2 v^k_j(x^k) + \frac{\nabla\tilde{G}^k_j(x^k)\nabla\tilde{G}^k_j(x^k)^\top}{\|\tilde{G}^k_j(x^k)\|} - \frac{\nabla\tilde{G}^k_j(x^k)\tilde{G}^k_j(x^k)\tilde{G}^k_j(x^k)^\top\nabla\tilde{G}^k_j(x^k)^\top}{\|\tilde{G}^k_j(x^k)\|^3}. \tag{3.9}$$

Notice that the right-hand side of the above formula can be evaluated since we have $G^k_j(x^k)$, $\nabla G^k_j(x^k)$ and $\nabla^2 v^k_j(x^k)$ by using (3.3), (3.4) and Proposition 2.2(d) with $\bar{x}$ replaced by x^k, respectively. Thus, we can calculate W_{kj} from (3.9). In the subsequent section, we will show quadratic convergence of Algorithm 1 in which B_k are chosen as (3.8).

Another plausible choice of B_k is to let $B_k = \nabla^2_{xx}\mathcal{L}^k_\varepsilon(x^k, \eta^k)$ for each k, where $\mathcal{L}^k_\varepsilon(\cdot,\cdot)$ denotes the Lagrangian of SOCP(x^k, ε). However, to evaluate $\nabla^2_{xx}\mathcal{L}^k_\varepsilon(x^k, \eta^k)$, we have to compute $\nabla^2 t^k_j(x^k)$ $(j = 1, 2, \ldots, r_\varepsilon(x^k))$, and it often brings about some numerical difficulties. On the other hand, computing (3.8) does not require any calculation of $\nabla^2 t^k_j(x^k)$.

4 Convergence analysis

In this section, we study global and local convergence properties of the proposed algorithm.

4.1 Global convergence

To begin with, we make the following assumption:

Assumption 4.1. *For each k,*

(a) x^k *is regular,*

(b) $\tilde{g}(x^k,t) \neq 0$ *for all* $t \in T_\varepsilon(x^k)$*,*

(c) $\mathrm{QSOCP}(x^k,\varepsilon)$ *is feasible, and the KKT conditions* (3.6) *hold at the unique optimum of* $\mathrm{QSOCP}(x^k,\varepsilon)$.

By Assumption 4.1 (a), SISOCP (1.1) can locally be reduced to $\mathrm{SOCP}(x^k,\varepsilon)$ around x^k for each k. By Assumption 4.1 (b), we can ensure the continuous differentiability of $\lambda(x^k,\cdot)$ at each $t \in T_\varepsilon(x^k)$, which is required by the regularity of x^k. Although Assumption 4.1 (b) may seem restrictive, $\tilde{g}(x^k,t)=0$ is unlikely to occur in practice at any local minimizer of $P(x^k)$, since $-\|\tilde{g}(x^k,\cdot)\|$ attains its "sharp" maximum at any $t \in T$ such that $\tilde{g}(x^k,t)=0$. Under Assumption 4.1 (c), $\mathrm{QSOCP}(x^k,\varepsilon)$ has a unique optimum d^k since $B_k \in S^n_{++}$.

By the following proposition, we can ensure that our algorithm finds a KKT point of SISOCP (1.1), when the termination criterion $d^k = 0$ is satisfied.

Proposition 4.2. *Suppose that Assumption* 4.1 *holds. If* $d^k=0$*, then the KKT conditions* (3.1) *and* (3.2) *for SISOCP* (1.1) *are satisfied at* x^k *with some Lagrange multiplier vectors* $\eta_1^{k+1}, \eta_2^{k+1}, \dots, \eta_{r_\varepsilon(x^k)}^{k+1} \in \mathbb{R}^m$*. In particular,* x^k *is feasible for SISOCP* (1.1).

Proof. From the regularity of x^k, $\bar{t}_j^k := t_j^k(x^k) \in T_\varepsilon(x^k)$ $(j=1,2,\dots,r_\varepsilon(x^k))$ are nondegenerate local optima of $P(x^k)$ and then satisfy the KKT conditions of the lower-level problem $P(x^k)$. Thus, we have, for each $j=1,2,\dots,r_\varepsilon(x^k)$,

$$\nabla_t g_1(x^k,\bar{t}_j^k) - \frac{\nabla_t \tilde{g}(x^k,\bar{t}_j^k)\tilde{g}(x^k,\bar{t}_j^k)}{\|\tilde{g}(x^k,\bar{t}_j^k)\|} - \sum_{i \in I_a(\bar{t}_j^k)} \alpha_i^j \nabla h_i(\bar{t}_j^k) = 0, \tag{4.1}$$

where $I_a(\bar{t}_j^k)$ is defined by (2.5) and α_i^j $(i \in I_a(\bar{t}_j^k))$ are Lagrange multipliers. Using the fact that $\nabla t_j^k(x^k)\nabla h_i(\bar{t}_j^k)=0$ holds for each $i \in I_a(\bar{t}_j^k)$ by Proposition 2.2 (c), (4.1) yields

$$\nabla t_j^k(x^k)\nabla_t g_1(x^k,\bar{t}_j^k) - \frac{\nabla t_j^k(x^k)\nabla_t \tilde{g}(x^k,\bar{t}_j^k)\tilde{g}(x^k,\bar{t}_j^k)}{\|\tilde{g}(x^k,\bar{t}_j^k)\|} = 0, \quad j=1,2,\dots,r_\varepsilon(x^k). \tag{4.2}$$

From the KKT conditions (3.6) of $\mathrm{QSOCP}(x^k,\varepsilon)$ with $d^k=0$ and (3.4), we obtain

$$\nabla f(x^k) - \sum_{j=1}^{r_\varepsilon(x^k)} \left(\nabla_x g(x^k,\bar{t}_j^k) + \nabla t_j^k(x^k)\nabla_t g(x^k,\bar{t}_j^k)\right)\eta_j^{k+1} = 0, \tag{4.3}$$

$$\mathcal{K}^m \ni \eta_j^{k+1} \perp g(x^k,\bar{t}_j^k) \in \mathcal{K}^m \ (j=1,2,\dots,r_\varepsilon(x^k)). \tag{4.4}$$

Notice that (4.4) implies $\eta_j^{k+1} = (\eta_j^{k+1})_1(1, -\tilde{g}(x^k,\bar{t}_j^k)^\top/\|\tilde{g}(x^k,\bar{t}_j^k)\|)^\top$ since $\|\tilde{g}(x^k,\bar{t}_j^k)\| \neq 0$ by Assumption 4.1 (b), which together with (4.2) yields

$$\nabla t_j^k(x^k)\nabla_t g(x^k,\bar{t}_j^k)\eta_j^{k+1} = 0, \quad j=1,2,\dots,r_\varepsilon(x^k).$$

Hence, from (4.3), we have $\nabla f(x^k) - \sum_{j=1}^{r_\varepsilon(x^k)} \nabla_x g(x^k,\bar{t}_j^k)\eta_j^{k+1} = 0$. Combining this and (4.4), we obtain the desired result.

The feasibility of x^k readily follows, since we have $g(x^k,\bar{t}_j^k) \in \mathcal{K}^m$ $(j=1,2,\dots,r_\varepsilon(x^k))$ from (4.4) and $\bar{t}_j^k$ $(j=1,2,\dots,r_\varepsilon(x^k))$ contain global optima of $P(x^k)$. □

We next show that the search direction $d^k \in \mathbb{R}^n$ obtained from $\mathrm{QSOCP}(x^k,\varepsilon)$ is a descent direction for $\Phi_\rho(\cdot)$ at x^k as long as the penalty parameter ρ is sufficiently large, which ensures that the line search in Step 4 terminates finitely at each iteration. To this end, we begin with proving the following lemma.

Lemma 4.3. *Suppose that Assumption* 4.1 *holds. Then, we have*

$$\varphi_+(x^k) + \varphi_+'(x^k; d^k) \le 0, \tag{4.5}$$

where $\varphi(\cdot)$ *is defined by* (2.2).

Proof. By the regularity of x^k and Proposition 2.4, there exist an open neighborhood $U_\varepsilon(x^k)$ of x^k and C^2 functions $t_j^k(\cdot) : U_\varepsilon(x^k) \to T$ $(j = 1, 2, \dots, r_\varepsilon(x^k))$ such that

$$\varphi(x) = \max_{t \in T}(-\lambda(x, t)) = \max_{j=1,2,\cdots,r_\varepsilon(x^k)}(-\lambda(x, t_j^k(x)) = \max_{j=1,2,\cdots,r_\varepsilon(x^k)}\left(\|\tilde{G}_j^k(x)\| - G_{j1}^k(x)\right)$$

for all $x \in U_\varepsilon(x^k)$, where $G_j^k(x) = (G_{j1}^k(x), \tilde{G}_j^k(x)) := (g_1(x, t_j^k(x)), \tilde{g}(x, t_j^k(x)))$ for $j = 1, 2, \dots, r_\varepsilon(x^k)$. Then, by letting $J(x^k) := \left\{j \in \{1, 2, \dots, r_\varepsilon(x^k)\} \mid -\lambda(x^k, t_j^k(x^k)) = \varphi(x^k)\right\}$, we have

$$\varphi(x^k) = -\lambda(x^k, t_j^k(x^k)) = \|\tilde{G}_j^k(x^k)\| - G_{j1}^k(x^k) \quad (j \in J(x^k)). \tag{4.6}$$

In addition, since $\tilde{G}_j^k(x^k) \ne 0$ from Assumption 4.1 (b), it is not difficult to show

$$\varphi_+'(x^k; d^k) = \begin{cases} 0 & \text{if } \varphi(x^k) < 0, \\ \displaystyle\max_{j \in J(x^k)} \left(\frac{\tilde{G}_j^k(x^k)^\top \nabla \tilde{G}_j^k(x^k)^\top d^k}{\|\tilde{G}_j^k(x)\|} - \nabla G_{j1}^k(x^k)^\top d^k\right)_+ & \text{if } \varphi(x^k) = 0, \\ \displaystyle\max_{j \in J(x^k)} \left(\frac{\tilde{G}_j^k(x^k)^\top \nabla \tilde{G}_j^k(x^k)^\top d^k}{\|\tilde{G}_j^k(x)\|} - \nabla G_{j1}^k(x^k)^\top d^k\right) & \text{if } \varphi(x^k) > 0. \end{cases} \tag{4.7}$$

Moreover, since d^k is feasible for QSOCP(x^k, ε), we have $G_j^k(x^k) + \nabla G_j^k(x^k)^\top d^k \in \mathcal{K}^m$ $(j = 1, 2, \dots, r_\varepsilon(x^k))$, which implies that

$$G_{j1}^k(x^k) + \nabla G_{j1}^k(x^k)^\top d^k - \left\|\tilde{G}_j^k(x^k) + \nabla \tilde{G}_j^k(x^k)^\top d^k\right\| \ge 0 \qquad (j = 1, 2, \dots, r_\varepsilon(x^k)). \tag{4.8}$$

Notice that, for any $u \in \mathbb{R}^n$ and $v \in \mathbb{R}^n$,

$$\|u\| + \frac{u^\top v}{\|u\|} \le \|u + v\| \tag{4.9}$$

holds since

$$\|u+v\|^2 - \left(\|u\| + \frac{u^\top v}{\|u\|}\right)^2 = \|v\|^2 - \frac{(u^\top v)^2}{\|u\|^2} \ge \|v\|^2 - \frac{\|u\|^2\|v\|^2}{\|u\|^2} = 0.$$

Hence, by setting $u := \tilde{G}_j^k(x), v := \nabla \tilde{G}_j^k(x^k)^\top d^k$ in (4.9), we have

$$\frac{\tilde{G}_j^k(x^k)^\top \nabla \tilde{G}_j^k(x^k)^\top d^k}{\|\tilde{G}_j^k(x)\|} + \|\tilde{G}_j^k(x^k)\| \le \left\|\tilde{G}_j^k(x^k) + \nabla \tilde{G}_j^k(x^k)^\top d^k\right\| \quad (j \in J(x^k)). \tag{4.10}$$

To show (4.5), we consider three cases (i) $\varphi(x^k) < 0$, (ii) $\varphi(x^k) = 0$, (iii) $\varphi(x^k) > 0$. In case (i), (4.7) implies $\varphi_+(x^k) + \varphi_+'(x^k; d^k) = \varphi_+(x^k) = 0$. In case (ii), since $\|\tilde{G}_j^k(x^k)\| - G_{j1}^k(x^k) = -\varphi(x^k) = 0$, it holds that, for $j \in J(x^k)$,

$$\begin{aligned} &\frac{\tilde{G}_j^k(x^k)^\top \nabla \tilde{G}_j^k(x^k)^\top d^k}{\|\tilde{G}_j^k(x)\|} - \nabla G_{j1}^k(x^k)^\top d^k \\ &= \|\tilde{G}_j^k(x^k)\| - G_{j1}^k(x^k) + \frac{\tilde{G}_j^k(x^k)^\top \nabla \tilde{G}_j^k(x^k)^\top d^k}{\|\tilde{G}_j^k(x)\|} - \nabla G_{j1}^k(x^k)^\top d^k \\ &\le \left\|\tilde{G}_j^k(x^k) + \nabla \tilde{G}_j^k(x^k)^\top d^k\right\| - G_{j1}^k(x^k) - \nabla G_{j1}^k(x^k)^\top d^k \\ &\le 0, \end{aligned} \tag{4.11}$$

where the first inequality follows from (4.10) and the second inequality does from (4.8). Then, we obtain from (4.7) and (4.11)

$$\varphi_+(x^k) + \varphi'_+(x^k; d^k) = \max_{j\in J(x^k)} \left(\frac{\tilde{G}_j^k(x^k)^\top \nabla \tilde{G}_j^k(x^k)^\top d^k}{\|\tilde{G}_j^k(x)\|} - \nabla G_{j1}^k(x^k)^\top d^k \right)_+ = 0.$$

In case (iii), since $\varphi_+(x^k) = \varphi(x^k)$, we have

$$\begin{aligned}
\varphi_+(x^k) + \varphi'_+(x^k; d^k) &= \varphi(x^k) + \max_{j\in J(x^k)} \left(\frac{\tilde{G}_j^k(x^k)^\top \nabla \tilde{G}_j^k(x^k)^\top d^k}{\|\tilde{G}_j^k(x)\|} - \nabla G_{j1}^k(x^k)^\top d^k \right) \\
&= \max_{j\in J(x^k)} \left(\frac{\tilde{G}_j^k(x^k)^\top \nabla \tilde{G}_j^k(x^k)^\top d^k}{\|\tilde{G}_j^k(x)\|} + \varphi(x^k) - \nabla G_{j1}^k(x^k)^\top d^k \right) \\
&= \max_{j\in J(x^k)} \left(\frac{\tilde{G}_j^k(x^k)^\top \nabla \tilde{G}_j^k(x^k)^\top d^k}{\|\tilde{G}_j^k(x)\|} + \|\tilde{G}_j^k(x^k)\| - G_{j1}^k(x^k) - \nabla G_{j1}^k(x^k)^\top d^k \right) \\
&\le \max_{j\in J(x^k)} \left(\left\| \tilde{G}_j^k(x^k) + \nabla \tilde{G}_j^k(x^k)^\top d^k \right\| - G_{j1}^k(x^k) - \nabla G_{j1}^k(x^k)^\top d^k \right) \\
&\le 0,
\end{aligned}$$

where the first and third equalities are obtained from (4.7) and (4.6), respectively, the first inequality follows from (4.10), and the last inequality is derived from (4.8). Consequently, we have the desired result. $\square$

Proposition 4.4. *Suppose that Assumption* 4.1 *holds. If* $\rho \ge \sum_{j=1}^{r_\varepsilon(x^k)} (\eta_j^{k+1})_1$, *then we have*

$$\Phi'_\rho(x^k; d^k) \le -(d^k)^\top B_k d^k. \tag{4.12}$$

Furthermore, $\Phi'_\rho(x^k; d^k) < 0$ *holds when* $d^k \neq 0$.

Proof. First note that, for each $j = 1, 2, \ldots, r_\varepsilon(x^k)$, we have

$$\begin{aligned}
G_j^k(x^k)^\top \eta_j^{k+1} &= (\eta_j^{k+1})_1 (G_j^k(x^k))_1 + (\tilde{\eta}_j^{k+1})^\top \tilde{G}_j^k(x^k) \\
&\ge (\eta_j^{k+1})_1 (G_j^k(x^k))_1 - \|\tilde{\eta}_j^{k+1}\| \|\tilde{G}_j^k(x^k)\| \\
&\ge (\eta_j^{k+1})_1 (G_j^k(x^k))_1 - (\eta_j^{k+1})_1 \|\tilde{G}_j^k(x^k)\| \\
&= (\eta_j^{k+1})_1 \lambda(x^k, t_j^k(x^k)) \\
&\ge (\eta_j^{k+1})_1 \min_{t\in T} \lambda(x^k, t), \\
&\ge -(\eta_j^{k+1})_1 \varphi_+(x^k),
\end{aligned} \tag{4.13}$$

where the second and third inequalities hold since $(\eta_j^{k+1})_1 \ge \|\tilde{\eta}_j^k\|$ by $\eta_j^{k+1} \in \mathcal{K}^m$, and the last inequality follows from $\varphi_+(x^k) \ge \varphi(x^k) = -\min_{t\in T} \lambda(x^k, t)$. Then, by noting the KKT conditions (3.5) and (3.6) of QSOCP(x^k, ε), we obtain

$$\begin{aligned}
\nabla f(x^k)^\top d^k &= -(d^k)^\top B_k d^k + \sum_{j=1}^{r_\varepsilon(x^k)} \left(\nabla G_j^k(x^k) \eta_j^{k+1} \right)^\top d^k \\
&= -(d^k)^\top B_k d^k - \sum_{j=1}^{r_\varepsilon(x^k)} G_j^k(x^k)^\top \eta_j^{k+1}, \\
&\le -(d^k)^\top B_k d^k + \sum_{j=1}^{r_\varepsilon(x^k)} (\eta_j^{k+1})_1 \varphi_+(x^k) \\
&\le -(d^k)^\top B_k d^k + \rho \varphi_+(x^k),
\end{aligned} \tag{4.14}$$

where the first equality holds since $(\eta_j^{k+1})^\top (G_j^k(x^k)+\nabla G_j^k(x^k)^\top d^k) = 0$ for each $j = 1, 2, \ldots, r_\varepsilon(x^k)$ by the SOC complementarity conditions in (3.6), the first inequality is derived from (4.13), and the last inequality is implied by $\rho \geq \sum_{j=1}^{r_\varepsilon(x^k)} (\eta_j^{k+1})_1 \geq 0$ and $\varphi_+(x^k) \geq 0$. By using these facts, we have

$$\begin{aligned}\Phi_\rho'(x^k; d^k) &= \nabla f(x^k)^\top d^k + \rho\varphi_+'(x^k; d^k) \\ &\leq -(d^k)^\top B_k d^k + \rho(\varphi_+(x^k) + \varphi_+'(x^k; d^k)) \\ &\leq -(d^k)^\top B_k d^k,\end{aligned}$$

where the first inequality follows from (4.14) and the last inequality does from Lemma 4.3. Therefore, (4.12) holds.

The latter claim is obvious from (4.12), $d^k \neq 0$ and $B_k \in S_{++}^n$. □

Assumption 4.5. (a) *There exist* $0 < \gamma_1 \leq \gamma_2$ *such that* $\gamma_1 \|d\|^2 \leq d^\top B_k d \leq \gamma_2 \|d\|^2$ *for all* $d \in \mathbb{R}^n$ *and* $k = 0, 1, 2, \ldots,$

(b) $\{x^k\}$ *is bounded, and*

(c) $\{d^k\}$ *is bounded.*

For an arbitrary accumulation point $x^* \in \mathbb{R}^n$ *of* $\{x^k\}$*, it holds that*

(d) x^* *is regular, and*

(e) $\tilde{g}(x^*, \bar{t}) \neq 0$ *for any* $\bar{t} \in T_\varepsilon(x^*)$*.*

Furthermore, let $U_\varepsilon(x^*) \subseteq \mathbb{R}^n$ *and* $t_j(\cdot) : U_\varepsilon(x^*) \to T$ $(j = 1, 2, \ldots, r_\varepsilon(x^*))$ *be an open neighborhood of* x^* *and functions, respectively, such that the conditions* (a)-(c) *in Proposition* 2.4 *hold with* $\bar{x}$ *replaced by* x^**. Then,*

(f) *there exists an open neighborhood* $V_\varepsilon(x^*) (\subseteq U_\varepsilon(x^*))$ *of* x^* *such that* $\{t_j(x)\}_{j=1}^{r_\varepsilon(x^*)} = T_\varepsilon(x)$ *holds for any* $x \in V(x^*)$*, and*

(g) *Slater's constraint qualification holds for* QSOCP(x^*, ε)*, i.e., there exists* $d_0 \in \mathbb{R}^n$ *such that* $G_j(x^*) + \nabla G_j(x^*)^\top d_0 \in \operatorname{int} \mathcal{K}^m$ *for all* $j = 1, 2, \ldots, r_\varepsilon(x^*)$*, where* $G_j(x) := g(x, t_j(x))$ $(j = 1, 2, \ldots, r_\varepsilon(x^*))$*.*

Assumption 4.5 (f) implies that, when $x^k \in V_\varepsilon(x^*)$, we have $G_j^k(x) \equiv G_j(x) := g(x, t_j(x))$ $(j = 1, 2, \ldots, r_\varepsilon(x^*))$, and hence SISOCP (1.1) can locally be reduced to the following SOCP around x^k:

$$\min_{x \in V_\varepsilon(x^*)} \ f(x) \quad \text{s.t. } G_j(x) \in \mathcal{K}^m, \ j = 1, 2, \ldots, r_\varepsilon(x^*). \tag{4.15}$$

In other words, the constraint functions of SOCP(x^k, ε) coincide with those of SOCP(x^*, ε), whenever $x^k \in V_\varepsilon(x^*)$. Note that Assumption 4.5 (f) holds when $T_\varepsilon(x)$ contains all local optima of the lower-level problem $P(x)$.

Under the above assumptions, we have the following proposition:

Proposition 4.6. *Suppose that Assumptions* 4.1 *and* 4.5 *hold. Let* $\eta_1^{k+1}, \eta_2^{k+1}, \ldots, \eta_{r_\varepsilon(x^k)}^{k+1} \in \mathcal{K}^m$ *be Lagrange multiplier vectors satisfying the KKT conditions* (3.5) *and* (3.6)*, and denote* $\eta^k := (\eta_1^k, \eta_2^k, \ldots, \eta_{r_\varepsilon(x^k)}^k)$*. Then, it holds that*

(a) $\{\eta^k\}$ *is bounded, and*

(b) *there exist some* $k_0 \geq 0$ *and* $\bar{\rho} > 0$ *such that* $\rho_k = \bar{\rho}$ *for all* $k \geq k_0$*.*

Proof. We first show (a). For contradiction, suppose that $\{\eta^k\}$ is not bounded. Then, there exists some subsequence $\{\eta^{k+1}\}_{k\in K}$ such that $\lim_{k\in K,\, k\to\infty}\|\eta^{k+1}\|=\infty$. We may assume, without loss of generality, that $\eta^{k+1}\neq 0$ for all $k\in K$. By Assumption 4.5(b), $\{x^k\}_{k\in K}$ is bounded and has at least one accumulation point, say, $x^*\in\mathbb{R}^n$. Again, without loss of generality, we can assume that $\lim_{k\in K,\, k\to\infty} x^k = x^*$. From Assumption 4.5(d), x^* is regular. Then, by Proposition 2.4, there exist some open neighborhood $U_\varepsilon(x^*)\subseteq\mathbb{R}^n$ of x^* and C^2 functions $t_j(\cdot):U_\varepsilon(x^*)\to T$ $(j=1,2,\ldots,r_\varepsilon(x^*))$ such that SISOCP (1.1) can locally be reduced to SOCP(x^*,ε) around x^*:

$$\min_{x\in U_\varepsilon(x^*)} f(x) \quad \text{s.t. } G_j(x)\in\mathcal{K}^m \ (j=1,2,\ldots,r_\varepsilon(x^*))$$

where $G_j(x):=g(x,t_j(x))$ $(j=1,2,\ldots,r_\varepsilon(x^*))$. From Assumption 4.5 (f), the constraint functions of SOCP(x^k,ε) for $k(\in K)\geq\bar{k}$ are identical to those of SOCP(x^*,ε) for some $\bar{k}\in K$ large enough. Therefore, QSOCP(x^k,ε) can be represented as

$$\min_{d\in\mathbb{R}^n} \frac{1}{2}d^\top B_k d+\nabla f(x^k)^\top d \quad \text{s.t. } G_j(x^k)+\nabla G_j(x^k)^\top d\in\mathcal{K}^m,\ j=1,2,\ldots,r_\varepsilon(x^*),$$

and its optimum d^k satisfies the following KKT conditions:

$$\begin{aligned}
&\nabla f(x^k)+B_k d^k-\sum_{j=1}^{r_\varepsilon(x^*)}\nabla G_j(x^k)\eta_j^{k+1}=0,\\
&\mathcal{K}^m\ni\eta_j^{k+1}\perp G_j(x^k)+\nabla G_j(x^k)^\top d^k\in\mathcal{K}^m \ (j=1,2,\ldots,r_\varepsilon(x^*)),
\end{aligned}$$

from which it follows that

$$\frac{1}{\|\eta^{k+1}\|}\nabla f(x^k)+\frac{B_k d^k}{\|\eta^{k+1}\|}-\sum_{j=1}^{r_\varepsilon(x^*)}\frac{\nabla G_j(x^k)\eta_j^{k+1}}{\|\eta^{k+1}\|}=0, \tag{4.16}$$

$$\mathcal{K}^m\ni\frac{\eta_j^{k+1}}{\|\eta^{k+1}\|}\perp G_j(x^k)+\nabla G_j(x^k)^\top d^k\in\mathcal{K}^m \ (j=1,2,\ldots,r_\varepsilon(x^*)) \tag{4.17}$$

for all $k(\in K)\geq\bar{k}$.

Note that $\{d^k\}_{k\in K}\subseteq\mathbb{R}^n$ is bounded from Assumption 4.5(c) and $\{\eta^{k+1}/\|\eta^{k+1}\|\}_{k(\in K)\geq\bar{k}}\subseteq\mathbb{R}^{mr_\varepsilon(x^*)}$ is also bounded. Let $(d^*,\eta^*):=(d^*,\eta_1^*,\eta_2^*,\ldots,\eta_{r_\varepsilon(x^*)}^*)\in\mathbb{R}^n\times\mathcal{K}^m\times\mathcal{K}^m\times\cdots\times\mathcal{K}^m$ be an arbitrary accumulation point of $\{(d^k,\eta^{k+1}/\|\eta^{k+1}\|)\}_{k(\in K)\geq\bar{k}}$. Without loss of generality, we can assume that

$$\lim_{k\in K,k\to\infty}\left(\frac{\eta^{k+1}}{\|\eta^{k+1}\|},d^k,x^k\right)=(\eta^*,d^*,x^*)\,. \tag{4.18}$$

Then, letting $k\in K, k\to\infty$ in (4.16) and (4.17) yields

$$\sum_{j=1}^{r_\varepsilon(x^*)}\nabla G_j(x^*)\eta_j^*=0, \tag{4.19}$$

$$\mathcal{K}^m\ni\eta_j^*\perp G_j(x^*)+\nabla G_j(x^*)^\top d^*\in\mathcal{K}^m \ (j=1,2,\ldots,r_\varepsilon(x^*)), \tag{4.20}$$

since $\{B_k\}$ is bounded from Assumption 4.5(a). Furthermore, from Assumption 4.5(g), QSOCP(x^*,ε) satisfies Slater's constraint qualification, i.e., there exists some $d_0\in\mathbb{R}^n$ such that

$$G_j(x^*)+\nabla G_j(x^*)^\top d_0\in\operatorname{int}\mathcal{K}^m \ (j=1,2,\ldots,r_\varepsilon(x^*)). \tag{4.21}$$

Now observe that

$$\sum_{j=1}^{r_\varepsilon(x^*)}(\eta_j^*)^\top\left(G_j(x^*)+\nabla G_j(x^*)^\top d_0\right)=\sum_{j=1}^{r_\varepsilon(x^*)}\left(\nabla G_j(x^*)\eta_j^*\right)^\top(d_0-d^*)=0, \tag{4.22}$$

where the first equality holds since $(\eta_j^*)^\top \left(G_j(x^*) + \nabla G_j(x^*)^\top d^*\right) = 0$ $(j = 1, 2, \ldots, r_\varepsilon(x^*))$ by (4.20), and the second equality follows from (4.19). Combining (4.22) with (4.21) and $\eta_j^* \in \mathcal{K}^m$ $(j = 1, 2, \ldots, r_\varepsilon(x^*))$, we obtain $\eta_j^* = 0$ for $j = 1, 2, \ldots, r_\varepsilon(x^*)$. This is a contradiction since $\|\eta^*\| = 1$ from (4.18). Therefore, $\{\eta^k\}$ is bounded.

We next show (b). For contradiction, we suppose that such $\bar{\rho} > 0$ and $k_0 \geq 0$ do not exist. Then, by the update rule in Step 3 of Algorithm 1, there exists an infinite subsequence $\{\rho_k\}_{k \in K'}$ of penalty parameters such that, for all $k \in K'$,

$$\rho_{k-1} < \sum_{j=1}^{r_\varepsilon(x^*)} (\eta_j^{k+1})_1 \quad \text{and} \quad \rho_k = \sum_{j=1}^{r_\varepsilon(x^*)} (\eta_j^{k+1})_1 + \delta,$$

from which we have $\rho_k \geq \rho_{k-1} + \delta$ for all $k \in K'$. This implies $\lim_{k\to\infty} \rho_k = \infty$, since $\{\rho_k\}$ is nondecreasing by the update rule. We thus obtain $\lim_{k\in K', k\to\infty} \|\eta^{k+1}\| = \infty$ since $\sum_{j=1}^{r_\varepsilon(x^k)} (\eta_j^k)_1 > \rho_{k-1} \to \infty$ as $k(\in K') \to \infty$. This contradicts the boundedness of $\{\eta^k\}$. $\square$

Now, we establish the global convergence of Algorithm 1.

Theorem 4.7. *Suppose that Assumptions* 4.1 *and* 4.5 *hold. Let $x^* \in \mathbb{R}^n$ be an arbitrary accumulation point of $\{x^k\} \subseteq \mathbb{R}^n$. Then, the KKT conditions* (3.1) *and* (3.2) *of SISOCP* (1.1) *hold at x^*.*

Proof. First, from Proposition 4.6, there exists some $\bar{\rho} > 0$ such that $\rho_k = \bar{\rho}$ for all k sufficiently large. For simplicity of expression, we assume that $\rho_k = \bar{\rho}$ for all k.

Choose a subsequence $\{x^k\}_{k\in K}$ such that $\lim_{k\in K, k\to\infty} x^k = x^*$. Since $\{d^k\}_{k\in K}$ is bounded from Assumption 4.5 (c), it has at least one accumulation point, say, $d^* \in \mathbb{R}^n$. To prove the desired result, from Proposition 4.2, it suffices to show $d^* = 0$. Due to Assumption 4.5 (d), (e) and (f), SISOCP (1.1) can locally be reduced to SOCP (4.15) around x^k for all $k \in K$ sufficiently large. Then, using the facts that $\{B_k\} \subseteq S_{++}^n$ is uniformly bounded by Assumption 4.5 (a), d^k is a descent direction of $\Phi_{\bar{\rho}}(\cdot)$ at x^k by Proposition 4.4 and $\Phi_{\bar{\rho}}(\cdot)$ is continuous everywhere, we can deduce that $d^* = 0$ in a way similar to the convergence analysis for the SQP-type method for solving the NLSOCP [11]. $\square$

4.2 Local Convergence

Now, we analyze the convergence rate of Algorithm 1. In the remainder of this section, we assume that a sequence $\{(x^k, \eta^k)\}$ generated by Algorithm 1 converges to $(x^*, \eta^*) \in \mathbb{R}^n \times \mathbb{R}^{mr_\varepsilon(x^*)}$. Moreover, we let $x^* \in \mathbb{R}^n$ be a regular point such that SISOCP (1.1) is locally reduced to the following SOCP(x^*, ε) around x^*:

$$\min_{x\in U_\varepsilon(x^*)} \; f(x) \;\text{ s.t. }\; G_j(x) \in \mathcal{K}^m \;\; (j = 1, 2, \ldots, r_\varepsilon(x^*)),$$

where, for $j = 1, 2, \ldots, r_\varepsilon(x^*)$, $G_j(x) := g(x, t_j(x))$ with C^2 functions $t_j(\cdot) : U_\varepsilon(x^*) \to T$ and an open neighborhood $U_\varepsilon(x^*) \subseteq \mathbb{R}^n$ of x^* satisfying conditions (a)-(c) in Proposition 2.4. We suppose that $\tilde{G}_j(x^*) \neq 0$ $(j = 1, 2, \ldots, r_\varepsilon(x^*))$ and (x^*, η^*) satisfies the KKT conditions for SOCP(x^*, ε):

$$\nabla f(x^*) - \sum_{j=1}^{r_\varepsilon(x^*)} \nabla G_j(x^*)\eta_j^* = 0,$$

$$\mathcal{K}^m \ni G_j(x^*) \perp \eta_j^* \in \mathcal{K}^m \;\; (j = 1, 2, \ldots, r_\varepsilon(x^*)),$$

where $\eta^* := (\eta_1^*, \eta_2^*, \dots, \eta_{r_\varepsilon(x^*)}^*) \in \mathbb{R}^m \times \mathbb{R}^m \times \cdots \times \mathbb{R}^m$. Furthermore, we define the Lagrangian of SOCP(x^*, ε) by

$$\mathcal{L}_\varepsilon(x, \eta) := f(x) - \sum_{j=1}^{r_\varepsilon(x^*)} G_j(x)^\top \eta_j,$$

where $\eta := (\eta_1, \eta_2, \dots, \eta_{r_\varepsilon(x^*)}) \in \mathbb{R}^m \times \mathbb{R}^m \times \cdots \times \mathbb{R}^m$.

Before discussing the convergence rate of the algorithm, we recall the constraint nondegeneracy and second order sufficient condition for SOCP(x^*, ε). We say that $x^* \in \mathbb{R}^n$ is constraint nondegenerate [2, Definition 16] if

$$\nabla G_j(x^*)^\top \mathbb{R}^n + \operatorname{lin} T_{\mathcal{K}^m}(G_j(x^*)) = \mathbb{R}^m$$

holds for each $j = 1, 2, \dots, r_\varepsilon(x^*)$, where $T_{\mathcal{K}^m}(z)$ denotes the tangent cone of $\mathcal{K}^m$ at $z \in \mathcal{K}^m$ and $\operatorname{lin} T_{\mathcal{K}^m}(z)$ stands for the largest linear subspace contained by $T_{\mathcal{K}^m}(z)$. The second order sufficient condition (SOSC) for general NLSOCP is studied in [2, 11, 25]. Under the assumption $\tilde{G}_j(x^*) \neq 0$ $(j = 1, 2, \dots, r_\varepsilon(x^*))$, the SOSC can be simplified as follows: For all $d \in C_{\mathcal{K}^m}(x^*, \eta^*) \setminus \{0\}$,

$$d^\top \nabla_{xx}^2 \mathcal{L}_\varepsilon(x^*, \eta^*) d + d^\top \left(\sum_{j=1}^{r_\varepsilon(x^*)} \mathcal{H}_\varepsilon^j(x^*, \eta^*) \right) d > 0,$$

where

$$\mathcal{H}_\varepsilon^j(x^*, \eta^*) := \begin{cases} -\dfrac{(\eta_j^*)_1}{G_{j1}(x^*)} \nabla G_j(x^*) \begin{pmatrix} 1 & 0 \\ 0 & -I_{m-1} \end{pmatrix} \nabla G_j(x^*)^\top & \text{if } G_j(x^*) \in \operatorname{bd} \mathcal{K}^m \setminus \{0\}, \\ 0 & \text{otherwise}, \end{cases}$$

$$C_{\mathcal{K}^m}(x^*, \eta^*) := \left\{ d \in \mathbb{R}^n \,\middle|\, d^\top \nabla G_j(x^*) \eta_j^* = 0 \text{ for all } j \text{ such that } G_j(x^*) \in \operatorname{bd} \mathcal{K}^m \setminus \{0\} \right\}.$$

Under the above conditions, we can show that the sequence $\{(x^k, \eta^k)\}$ converges to (x^*, η^*) quadratically, by using an argument in [25, Theorem 4.2].

Proposition 4.8. *Let $B : \mathbb{R}^n \times \mathbb{R}^{r_\varepsilon(x^*)} \to \mathbb{R}^{n \times n}$ be a function such that $B(x^*, \eta^*) = \nabla_{xx} \mathcal{L}_\varepsilon(x^*, \eta^*)$ and $B(\cdot, \cdot)$ is continuously differentiable at (x^*, η^*). Suppose that Assumption* 4.1 *and Assumption* 4.5 (d)-(g) *hold. Moreover, let the constraint nondegeneracy condition and SOSC hold at (x^*, η^*). If (x^{k_0}, η^{k_0}) is sufficiently close to (x^*, η^*) for some $k_0 \geq 0$, and if $s_k = 1$, $B_k = B(x^k, \eta^k)$ and $B_k \in S_{++}^n$ for all $k \geq k_0$, then $\{(x^k, \eta^k)\}$ converges to (x^*, η^*) quadratically.*

Proof. From Assumption 4.5 (f)-(g), if x^k is sufficiently close to x^*, then we can locally reduce SISOCP (1.1) to SOCP(x^*, ε) around x^k. Then, by [25, Theorem 4.2], we obtain the desired result. □

Using the above theorem, we can establish quadratic convergence of Algorithm 1 in which B_k are chosen as (3.8).

Theorem 4.9. *Suppose that the assumptions in Proposition* 4.8 *hold. If (x^{k_0}, η^{k_0}) is sufficiently close to (x^*, η^*) for some $k_0 \geq 0$, and if $s_k = 1$, B_k is chosen as* (3.8) *with $\zeta_j^k := \eta_j^k$ $(j = 1, 2, \dots, r_\varepsilon(x^*))$ and $B_k \in S_{++}^n$ for all $k \geq k_0$, then $\{(x^k, \eta^k)\}$ converges to (x^*, η^*) quadratically.*

Proof. From Assumption 4.5 (f)-(g), if x^k is sufficiently close to x^*, then we can locally reduce SISOCP (1.1) to SOCP(x^*, ε) around x^k. Then, by letting

$$B(x, \eta) := \nabla^2 f(x) - \sum_{j=1}^{r_\varepsilon(x^*)} (\eta_j)_1 \left(\nabla^2 G_{j1}(x) - \frac{\nabla^2 \tilde{G}_j(x) \tilde{G}_j(x)}{\|\tilde{G}_j(x)\|} \right),$$

we have $B(x^k, \eta^k) = B_k$ for all k sufficiently large. Hence, from Proposition 4.8, we have only to show that $B(\cdot, \cdot)$ is continuously differentiable at (x^*, η^*) and $B(x^*, \eta^*) = \nabla^2_{xx}\mathcal{L}_\varepsilon(x^*, \eta^*)$. The first claim is obvious since $\tilde{G}_j(x^*) \neq 0$ $(j = 1, 2, \ldots, r_\varepsilon(x^*))$. We prove the second claim. Notice that $\eta_j^* = (\eta_j^*)_1(1, -\tilde{G}_j(x^*)^\top/\|\tilde{G}_j(x^*)\|)^\top$ for $j = 1, 2, \ldots, r_\varepsilon(x^*)$ since $\tilde{G}_j(x^*) \neq 0$ and $\mathcal{K}^m \ni \eta_j^* \perp G_j(x^*) \in \mathcal{K}^m$ for $j = 1, 2, \ldots, r_\varepsilon(x^*)$. Thus, we have

$$\begin{aligned}
\nabla^2_{xx}\mathcal{L}_\varepsilon(x^*, \eta^*) &= \nabla^2 f(x^*) - \sum_{j=1}^{r_\varepsilon(x^*)} \nabla^2 G_j(x^*)\eta_j^* \\
&= \nabla^2 f(x^*) - \sum_{j=1}^{r_\varepsilon(x^*)} (\eta_j^*)_1 \nabla^2 G_j(x^*) \begin{pmatrix} 1 \\ -\dfrac{\tilde{G}_j(x^*)}{\|\tilde{G}_j(x^*)\|} \end{pmatrix} \\
&= B(x^*, \eta^*).
\end{aligned}$$

This completes the proof. □

5 Numerical Experiments

In this section, we report some numerical results. The program was coded in Matlab 2008a and run on a machine with an Intel®Core2 Duo E6850 3.00GHz CPU and 4GB RAM. Throughout the experiments, we let the index set be given by $T := \{t \in \mathbb{R} \mid h(t) \geq 0\}$, where $h(t) := (t+1, 1-t)^\top$, i.e., $T = [-1, 1]$. The actual implementation of Algorithm 1 was carried out as follows. To obtain $T_\varepsilon(x^k)$, we compute local minimizers of the lower-level problem $P(x^k)$. For this purpose, we first compute $\lambda(x^k, t)$ for $t = -1, -0.98, -0.96, \ldots, 0.98, 1$, where $\lambda(\cdot, \cdot)$ is defined as in (2.1). We then find local minimizers among $\lambda(x^k, -1), \lambda(x^k, -0.98), \ldots, \lambda(x^k, 1)$ and apply Newton's method with them as starting points. In Step 0, we set the parameters as $\alpha = 0.5$, $\beta = 10^{-5}$, $\delta = 5$, $\varepsilon = 0.1$ and $\rho_{-1} = 10$. The initial point $x^0 \in \mathbb{R}^n$ and the initial matrix $B_0 \in \mathbb{R}^{n\times n}$ are chosen as $x^0 := (10, 10, 10, \ldots, 10)^\top$ and the identity matrix I_n, respectively. In Step 1, we make use of the smoothing method [4, 8] to solve QSOCP(x^k, ε). In Step 2, we stop the algorithm when $\|d^k\| \leq 10^{-7}$ is satisfied. In Step 5, we update the matrix $B_k \in \mathbb{R}^{n\times n}$ by (3.8), where the vectors ζ_j^k $(j = 1, 2, \ldots, r_\varepsilon(x^k))$ are set as

$$\zeta_j^k := \begin{cases} \eta_i^k & \text{if we find } i \in \{1, 2, \ldots, r_\varepsilon(x^k)\} \text{ such that } t_j^k(\cdot) = t_i^{k-1}(\cdot) \\ 0 & \text{otherwise.} \end{cases} \tag{5.1}$$

In (5.1), we regard $t_j^k(\cdot) = t_i^{k-1}(\cdot)$ when

$$j \in \operatorname{argmin}_{l=1,2,\ldots,r_\varepsilon(x^{k-1})} \|t_l^k(x^k) - t_i^{k-1}(x^{k-1}) - \nabla t_i^{k-1}(x^{k-1})^\top (x^k - x^{k-1})\|$$

and

$$\|t_j^k(x^k) - t_i^{k-1}(x^{k-1}) - \nabla t_i^{k-1}(x^{k-1})^\top (x^k - x^{k-1})\| \leq 10^{-4}.$$

Moreover, to ensure positive definiteness of B_k, we modify B_k as follows. Let $\alpha_{ki} \in \mathbb{R}$ $(i = 1, 2, \ldots, n)$ and $V_k \in \mathbb{R}^{n\times n}$ be scalars and a matrix such that $B_k = V_k \mathrm{diag}(\alpha_{ki})_{i=1}^n V_k^\top$, respectively. Then, for each i, we replace α_{ki} by 10^{-4} if $\alpha_{ki} \leq 10^{-5}$, and redefine B_k as $B_k = V_k \mathrm{diag}(\alpha_{ki})_{i=1}^n V_k^\top$.

Experiment 1

In the first experiment, we examine the convergence behavior of Algorithm 1 by solving the vector-valued Chebyshev approximation problem (1.3). Let $Q : \mathbb{R} \to \mathbb{R}^3$ and $q : \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}^3$ be defined by

$$Q(t) := \begin{pmatrix} e^{t^2} + \cos t^2 \\ 2te^{t^2} - 2t\sin t^2 \\ (4t^2+2)e^{t^2} - 2\sin t^2 - 4t^2\cos t^2 \end{pmatrix}, \ q(u,t) := \begin{pmatrix} \sum_{\nu=1}^{n} u_\nu t^{\nu-1} \\ \sum_{\nu=2}^{n} (\nu-1) u_\nu t^{\nu-2} \\ \sum_{\nu=3}^{n} (\nu-1)(\nu-2) u_\nu t^{\nu-3} \end{pmatrix}.$$

To find a $u \in \mathbb{R}^n$ such that $q(u,t) \approx Q(t)$ over $t \in T$, we solve the following problem:

$$\operatorname*{Minimize}_{u \in \mathbb{R}^n} \ \max_{t \in T} \ \|Q(t) - q(u,t)\|. \tag{5.2}$$

As in (1.3), by using an auxiliary variable $v \in \mathbb{R}$, we can reformulate (5.2) as the following SISOCP with the four-dimensional SOC:

$$\begin{aligned} &\operatorname*{Minimize}_{(v,u) \in \mathbb{R} \times \mathbb{R}^n} \quad v \\ &\text{subject to} \quad \begin{pmatrix} 1 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 1 & t & t^2 & \cdots & t^n \\ 0 & 0 & 1 & 2t & \cdots & nt^{n-1} \\ 0 & 0 & 0 & 2 & \cdots & n(n-1)t^{n-2} \end{pmatrix} \begin{pmatrix} v \\ u \end{pmatrix} - \begin{pmatrix} 0 \\ e^{t^2} + \cos t^2 \\ 2te^{t^2} - 2t\sin t^2 \\ (4t^2+2)e^{t^2} - 2\sin t^2 - 4t^2\cos t^2 \end{pmatrix} \in \mathcal{K}^4 \\ &\qquad\qquad \text{for all } t \in T. \end{aligned} \tag{5.3}$$

We then apply Algorithm 1 to SISOCP (5.3) with $n = 6$ and $n = 8$. The obtained results are shown in Tables 1 and 2, where cpu(s) denotes the running time of Algorithm 1 in seconds, and $\mathrm{KKT}(x^k, \eta^k)$ is given by

$$\mathrm{KKT}(x^k, \eta^k) := \begin{pmatrix} \nabla f(x^k) - \sum_{j=1}^{r_\varepsilon(x^k)} \nabla_x g(x^k, \bar{t}_j^k) \eta_i^k \\ \eta_1^k - \mathrm{Proj}_{\mathcal{K}^m}\left(\eta_1^k - g(x^k, \bar{t}_1^k)\right) \\ \vdots \\ \eta_{r_\varepsilon(x^k)}^k - \mathrm{Proj}_{\mathcal{K}^m}\left(\eta_{r_\varepsilon(x^k)}^k - g(x, \bar{t}_{r_\varepsilon(x^k)}^k)\right) \end{pmatrix}$$

with $\mathrm{Proj}_{\mathcal{K}^m}(z) := \operatorname{argmin}_{w \in \mathcal{K}^m} \|z - w\|$, and $\eta^k = (\eta_1^k, \ldots, \eta_{r_\varepsilon(x^k)}^k) \in \mathbb{R}^m \times \cdots \times \mathbb{R}^m$ is a Lagrange multiplier vector satisfying the KKT conditions (3.5) and (3.6) for $\mathrm{QSOCP}(x^k, \varepsilon)$. Note that $\mathrm{KKT}(x^k, \eta^k) = 0$ if and only if (x^k, η^k) satisfies the KKT conditions (3.1) and (3.2) for SISOCP (1.1). From the tables, we can observe that Algorithm 1 succeeds in getting an optimal solution for SISOCP (5.3). Indeed, x^k and η^k satisfy the KKT conditions for SISOCP (5.3) accurately, since $\|\mathrm{KKT}(x^k, \eta^k)\| \le 10^{-10}$ at the last iteration. Also, we can observe that the step size s_k equals 1 in the final stage and $\{x^k\}$ converges to a solution rapidly. In addition, we confirm that $|T_\varepsilon(x^k)|$ becomes constant and the implicit functions $\{t_j^k(\cdot)\}_{j=1}^{r_\varepsilon(x^k)}$ remain unchanged eventually, and hence Assumption 4.5 (f) holds.

k	s_k	$\|d^k\|$	$\|\mathrm{KKT}(x^k, \eta^k)\|$	$\|T_\varepsilon(x^k)\|$
1	1.0	1.73e+01	1.47e+02	1
2	0.5	1.47e+01	7.06e-01	2
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
6	1.0	9.18e-04	5.99e-04	5
7	1.0	4.92e-07	2.63e-07	5
8	1.0	7.83e-11	4.20e-11	5
cpu(s): 5.8 seconds				

Table 1: Results for Experiment 1 ($n = 6$)

k	s_k	$\|d^k\|$	$\|\mathrm{KKT}(x^k, \eta^k)\|$	$\|T_\varepsilon(x^k)\|$
1	1.0	2.23e+01	5.03e+02	1
2	0.5	1.46e+01	7.06e-01	1
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
10	0.5	7.94e-07	2.51e-06	7
11	1.0	3.97e-07	1.26e-06	7
12	1.0	1.78e-12	5.64e-12	7
cpu(s): 22.4 seconds				

Table 2: Results for Experiment 1 ($n = 8$)

Experiment 2

In Experiment 1, we have observed that Algorithm 1 obtains accurate solutions with a rapid convergence rate. Thus, if a starting point is chosen near an optimal solution, Algorithm 1 is expected to find a solution more efficiently. In this experiment, to produce such a starting point, we use the regularized explicit exchange method (REEM) from [16], and then use Algorithm 1 with an approximate solution computed by the REEM. The REEM was implemented as described in Experiment 3-1 of [16]. The computational results for SISOCP (5.3) with $n = 6$ and $n = 8$ are shown in Table 3 and Table 4, respectively, where

- cpu(s) (REEM): the running time of the REEM
- cpu(s) (Algorithm 1): the running time of Algorithm 1
- cpu(s) (REEM+Algorithm 1): the total running time of the REEM and Algorithm 1.

From the tables, we observe that the total computational times are much less than those in Experiment 1. In particular, when $n = 8$, Algorithm 1 combined with the REEM took only 15.3 seconds in total, while Algorithm 1 alone spent 22.4 seconds in Experiment 1.

cpu(s) (REEM)	2.0 seconds
cpu(s) (Algorithm 1)	0.49 seconds
cpu(s) (REEM+Algorithm 1)	2.49 seconds

Table 3: Results for Experiment 2 ($n = 6$)

cpu(s) (REEM)	4.0 seconds
cpu(s) (Algorithm 1)	11.3 seconds
cpu(s) (REEM+Algorithm 1)	15.3 seconds

Table 4: Results for Experiment 2 ($n = 8$)

Experiment 3

In the third experiment, we implemented another SQP-type algorithm, which is also expected to find accurate solutions rapidly, and compared it with Algorithm 1 by solving the following SISOCP that contains multiple SOCs:

$$\begin{aligned} \underset{x\in\mathbb{R}^{10}}{\text{Minimize}} \quad & \tfrac{1}{2}x^\top Mx + c^\top x \\ \text{subject to} \quad & A^s(t)x - b^s(t) \in \mathcal{K}^{m_s} \text{ for all } t \in T, \\ & s = 1, 2, \dots, S, \end{aligned} \tag{5.4}$$

where $c \in \mathbb{R}^{10}$, $A^s(t) := (A^s_{ij}(t)) \in \mathbb{R}^{m_s \times 10}$ with $A^s_{ij}(t) := \sum_{\ell=0}^5 \alpha^s_{ij\ell} t^\ell$ $(i = 1, 2, \dots, m_s, j = 1, 2, \dots, 10)$ and $b^s(t) := (b^s_i(t)) \in \mathbb{R}^{m_s}$ with $b^s_1(t) := -\sum_{i=2}^{m_s} \sum_{\ell=0}^5 |\beta^s_{i\ell}|$ and $b^s_i(t) := \sum_{\ell=0}^5 \beta^s_{i\ell} t^\ell$ $(i = 2, \dots, m_s)$. The SOCs $\mathcal{K} := \mathcal{K}^{m_1} \times \mathcal{K}^{m_2} \times \cdots \mathcal{K}^{m_S}$ are chosen as in Table 5. For each type of SOC $\mathcal{K}$, we generate 50 problems as follows: The problem data $\alpha^s_{1j\ell}, \alpha^s_{ij\ell}, \beta^s_{i\ell}$ $(i = 2, \dots, m_s,\ j = 1, 2, \dots, 10,\ \ell = 0, 1, 2, \dots, 5,\ s = 1, 2, \dots, S)$ are chosen randomly from the interval $[2, -2]$. All components of c are randomly chosen from the interval $[5, -5]$. The matrix M is set to be $M := M_1^\top M_1 + 0.1 I_n$, where $M_1 \in \mathbb{R}^{n\times n}$ is a matrix whose entries are randomly chosen from the interval $[1, -1]$. Notice that, by the choice of $b^s_1(t)$, we can ensure that (5.4) is feasible.[2] In Step 3, we use the following penalty function for SISOCP (5.4) with multiple SOCs, which is a natural extension of the function defined by (3.7):

$$\Phi_\rho(x) := f(x) + \rho \sum_{s=1}^S \varphi^s_+(x), \tag{5.5}$$

where $\varphi^s(x) := \max_{t\in T} \left(-A^s_1(t)x + b^s_1(t) + \|\tilde{A}^s(t)(x) - \tilde{b}^s(t)\|\right)$ for $s = 1, 2, \dots, S$. Accordingly, we extend the update rule of the penalty parameters $\{\rho_k\}$ in Step 5 as follows:

$$\rho_k := \begin{cases} \rho_{k-1} & \text{if } \rho_{k-1} \geq \max\limits_{s=1,2,\dots,S} \sum\limits_{j=1}^{r^s_\varepsilon(x^k)} (\eta^{k+1}_{sj})_1 \\ \delta + \max\limits_{s=1,2,\dots,S} \sum\limits_{j=1}^{r^s_\varepsilon(x^k)} (\eta^{k+1}_{sj})_1 & \text{otherwise,} \end{cases}$$

where $\delta > 0$ is a given constant and η^{k+1}_{sj} $(j = 1, 2, \dots, r^s_\varepsilon(x^k), s = 1, 2, \dots, S)$ are Lagrange multiplier vectors obtained by solving QSOCP(x^k, ε) for SISOCP (5.4).

We next explain another SQP-type algorithm, which we call the QP-based method. For simplicity of expression, we consider the case of SISOCP (1.1) with a single SOC. In the QP-based method, we reformulate SOCP(x^k, ε) as the following nonlinear program that does not contain SOC constraints explicitly:

$$\min_{x\in U(x^k)} f(x) \text{ s.t. } v^k_j(x) \geq 0 \ (j = 1, 2, \dots, r_\varepsilon(x^k)), \tag{5.6}$$

[2] The origin $x = 0$ always lies in the interior of the feasible region, since we have $-b^s(t) \in \operatorname{int} \mathcal{K}^{m_s}$ from $-b^s_1(t) - \|(-b^s_2(t), \dots, -b^s_{m_s}(t))^\top\| > 0$ for all $t \in T$.

where $v_j^k(x) := \lambda(x, t_j^k(x))$ for $j = 1, 2, \ldots, r_\varepsilon(x^k)$, and then generate a search direction d^k by solving the following Quadratic Program[3] (QP):

$$\mathrm{QP}(x^k, \varepsilon): \quad \begin{array}{ll} \text{Minimize} & \nabla f(x^k)^\top d + \frac{1}{2} d^\top \tilde{B}_k d \\ \text{subject to} & v_j^k(x^k) + \nabla v_j^k(x^k)^\top d \geq 0 \ (j = 1, 2, \ldots, r_\varepsilon(x^k)), \end{array}$$

where $\tilde{B}_k \in S^n_{++}$. We make use of the Hessian of the Lagrangian of (5.6). Specifically, we first compute

$$\tilde{D}_k := \nabla^2 f(x^k) - \sum_{j=1}^{r_\varepsilon(x^k)} \xi_j^k \nabla^2 v_j^k(x^k),$$

with

$$\xi_j^k := \begin{cases} \tilde{\xi}_i^k & \text{if we find } i \in \{1, 2, \ldots, r_\varepsilon(x^k)\} \text{ such that } t_j^k(\cdot) = t_i^{k-1}(\cdot) \\ 0 & \text{otherwise,} \end{cases}$$

where $\tilde{\xi}_i^k \in \mathbb{R}$ $(i = 1, 2, \ldots, r_\varepsilon(x^{k-1}))$ are Lagrange multipliers satisfying the KKT conditions of $\mathrm{QP}(x^{k-1}, \varepsilon)$. Note that $\nabla v_j^k(x^k)$ and $\nabla^2 v_j^k(x^k)$ for $j = 1, 2, \ldots, r_\varepsilon(x^k)$ can be calculated from Proposition 2.2 (d). Similarly to the matrix B_k for Algorithm 1, we also ensure the positive definiteness of $\tilde{B}_k$ as follows: Let $\tilde{\alpha}_{ki} \in \mathbb{R}$ $(i = 1, 2, \ldots, n)$ and $\tilde{V}_k \in \mathbb{R}^{n \times n}$ be scalars and a matrix such that $\tilde{B}_k = \tilde{V}_k \mathrm{diag}(\tilde{\alpha}_{ki})_{i=1}^n \tilde{V}_k^\top$, respectively. Then, for each i, we replace $\tilde{\alpha}_{ki}$ by 10^{-4} if $\tilde{\alpha}_{ki} \leq 10^{-5}$, and redefine $\tilde{B}_k$ as $\tilde{B}_k = \tilde{V}_k \mathrm{diag}(\tilde{\alpha}_{ki})_{i=1}^n \tilde{V}_k^\top$. We use the penalty function defined by (3.7) (by (5.5) for problem (5.4)) and determine a step size by the Armijo line search. We update the penalty parameters $\{\rho_k\}$ as follows: If $\rho_{k-1} \geq \sum_{j=1}^{r_\varepsilon(x^k)} \tilde{\xi}_j^{k+1}$, then we set $\rho_k := \rho_{k-1}$. Otherwise, we set $\rho_k := \sum_{j=1}^{r_\varepsilon(x^k)} \tilde{\xi}_j^{k+1} + \delta$, where $\delta > 0$ is a given constant.

We extend the above QP-based method to (5.4) and implement it. The choice of parameters in the QP-based method is the same as in Algorithm 1. Moreover, we solve $\mathrm{QP}(x^k, \varepsilon)$ with the solver *quadprog* in MATLAB Optimization Toolbox.

The obtained results are shown in Table 5, where each column represents the following:

- $\text{ite}_{\max}$: the maximum number of iterations among 50 problems for each $\mathcal{K}$
- $\text{ite}_{\min}$: the minimum number of iterations among 50 problems for each $\mathcal{K}$
- ite_{ave}: the average number of iterations over 50 problems for each $\mathcal{K}$
- cpu(s): the average time in seconds over 50 problems for each $\mathcal{K}$

For all the generated problems, both algorithms successfully obtain optimal solutions. From the table, we can observe that Algorithm 1 tends to perform better than the QP-based method. In particular, when $\mathcal{K} = 10$, both the number of iterations and the computational time for Algorithm 1 are less than half of those for the QP-based method. This fact suggests that Algorithm 1 may exploit the structure of SOC more effectively than the QP-based method.

6 Concluding Remarks

For solving the semi-infinite program with an infinite number of SOC constraints, we proposed the local reduction based SQP-type method. We studied the global and local convergence properties of the proposed algorithm. Finally, in the numerical experiments, we actually implemented and examined its effectiveness. For the sake of comparison, we also implemented another SQP-type method and observed good performance of the proposed algorithm.

[3] We also suppose that Assumption 4.1 (b) holds.

$\mathcal{K}$	Algorithm 1				QP-based method			
	$\text{ite}_{\max}$	$\text{ite}_{\min}$	ite_{ave}	cpu(s)	$\text{ite}_{\max}$	$\text{ite}_{\min}$	ite_{ave}	cpu(s)
$\mathcal{K}^{10}$	19	3	6.22	1.04	49	6	13.28	1.91
$\mathcal{K}^{30}$	12	3	5.34	2.06	41	6	12.52	4.09
$\mathcal{K}^{50}$	11	3	5.54	2.66	31	7	13.36	4.04
$\mathcal{K}^{20} \times \mathcal{K}^{30}$	17	3	5.56	2.95	24	7	11.48	4.55
$\mathcal{K}^{20} \times \mathcal{K}^{15} \times \mathcal{K}^{15}$	13	3	5.95	5.91	23	7	11.46	6.36

Table 5: Comparison of Algorithm 1 and the QP-based method (Experiment 3)

References

[1] Alizadeh, F., Goldfarb, D.: Second-order cone programming. Math. Program. **95**, 3–51 (2003)

[2] Bonnans, J.F., Ramírez, C.H.: Perturbation analysis of second-order cone programming problems. Math. Program. **104**, 205–227 (2005)

[3] Floudas, C.A., Stein, O.: The adaptive convexification algorithm: a feasible point method for semi-infinite programming. SIAM J. Optim. **18**, 1187–1208 (2007)

[4] Fukushima, M., Luo, Z.Q., Tseng, P.: Smoothing functions for second-order cone complementarity problems. SIAM J. Optim. **12**, 436–460 (2001)

[5] Goberna, M.A., López, M.A.: Semi-Infinite Programming—Recent Advances. Kluwer Academic Publishers, Dordrecht (2001)

[6] Gramlich, G., Hettich, R., Sachs, E.W.: Local convergence of SQP methods in semi-infinite programming. SIAM J. Optim. **5**, 641–658 (1995)

[7] Hayashi, S., Wu, S.-Y.: An explicit exchange algorithm for linear semi-infinite programming problems with second-order cone constrains. SIAM J. Optim. **20**, 1527–1546 (2009)

[8] 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)

[9] Hettich, R.: An implementation of a discretization method for semi-infinite programming. Math. Program. **34**, 354–361 (1986)

[10] Hettich, R., Kortanek, K.O.: Semi-infinite programming: Theory, methods, and applications. SIAM Rev. **35**(3), 380–429 (1993)

[11] Kato, H., Fukushima, M.: An SQP-type algorithm for nonlinear second-order cone programs. Optim. Lett. **1**, 129–144 (2007)

[12] Lai, H.C., Wu, S.-Y.: On linear semi-infinite programming problems. Numer. Funct, Anal. Optim. **13**, 287–304 (1992)

[13] Li, D.H., Qi, L., Tam, J., Wu, S.-Y.: A smoothing Newton method for semi-infinite programming. J. Global Optim. **30**, 169–194 (2004)

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

[15] López, M.A., Still, G.: Semi-infinite programming. European J. Oper. Res. **180**, 491–518 (2007)

[16] 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)

[17] Pereira, A., Costa, M., Fernandes, E.: Interior point filter method for semi-infinite programming problems. Optimization **60**, 1309–1338 (2011)

[18] Pereira, A., Fernandes, E.: A reduction method for semi-infinite programming by means of a global stochastic approach. Optimization **58**, 713–726 (2009)

[19] Qi, L., Wu, S.-Y., Zhou, G.: Semismooth Newton methods for solving semi-infinite programming problems. J. Global Optim. **27**, 215–232 (2003)

[20] Reemtsen, R.: Discretization methods for the solution of semi-infinite programming problems. J. Optim. Theory Appl. **71**, 85–103 (1991)

[21] Reemtsen, R., Rückmann, J.-J.(eds.): Semi-Infinite Programming. Kluwer Academic Publishers, Boston (1998)

[22] Sturm, J.F.: Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. Optim. Methods Softw. **11**, 625–653 (1999)

[23] Tanaka, Y., Fukushima, M., Ibaraki, T.: A globally convergent SQP method for semi-infinite nonlinear optimization. J. Comput. Appl. Math. **23**, 141–153 (1988)

[24] Toh, K.C., Todd, M.J., Tütüncü, R.H. (1999): SDPT3—a MATLAB software package for semidefinite programming, version 2.1. Optim. Methods Softw. **11**, 545–581 (1999)

[25] Wang, Y., Zhang, L.: Properties of equation reformulation of the Karush–Kuhn–Tucker condition for nonlinear second order cone optimization problems. Math. Meth. Oper. Res. **70**, 195–218 (2009)

[26] Wu, S.-Y., Li, D.H., Qi, L., Zhou, G.: An iterative method for solving KKT system of the semi-infinte programming. Optim. Methods Softw. **20**, 629–643 (2005)

[27] Yamashita, H., Yabe, H.: A primal-dual interior point method for nonlinear optimization over second-order cones. Optim. Methods Softw. **24**, 407–426 (2009)