数理解析研究所講究録
第 1726 巻 2011 年 161-171

Optimality Conditions and Algorithms for Semi-Infinite Programs with an Infinite Number of Second-Order Cone Constraints

奥野 貴之 (Takayuki Okuno)　林 俊介 (Shunsuke Hayashi)
福島 雅夫 (Masao Fukushima)
京都大学大学院 情報学研究科 数理工学専攻
Department of Applied Mathematics and Physics
Graduate School of Informatics, Kyoto University

Abstract

The semi-infinite program (SIP) is normally represented with infinitely many inequality constraints, and has been much studied so far. However, there have been very few studies on the SIP involving second-order cone (SOC) constraints, even though it has important applications such as Chebychev-like approximation and filter design.

In this paper, we focus on the SIP with a convex objective function and infinitely many SOC constraints, called the SISOCP for short. We show that, under a generalized Slater constraint qualification, an optimum of the SISOCP satisfies the KKT conditions that can be represented only with a *finite* subset of the SOC constraints. Next we introduce the regularization and the explicit exchange methods for solving the SISOCP. We first provide an explicit exchange method without a regularization technique, and show that it has global convergence under the strict convexity assumption on the objective function. Then we propose an algorithm combining a regularization method with the explicit exchange method. For the SISOCP, we establish global convergence of the hybrid algorithm without the strict convexity assumption.

1 Introduction

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

$$\begin{array}{ll} \text{Minimize} & f(x) \\ \text{subject to} & A(t)^\top x - b(t) \in \mathcal{K} \quad \text{for all } t \in T \end{array} \tag{1.1}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is a continuously differentiable convex function, $A : \mathbb{R}^\ell \to \mathbb{R}^{n\times m}$ and $b : \mathbb{R}^\ell \to \mathbb{R}^m$ are continuous functions, $T \subseteq \mathbb{R}^\ell$ is a given compact set, and $\mathcal{K}^{m_i} \subseteq \mathbb{R}^{m_i}$ $(i = 1, 2, \ldots, s)$ is the second-order cone (SOC), that is, $\mathcal{K} := \mathcal{K}^{m_1} \times \mathcal{K}^{m_2} \times \cdots \times \mathcal{K}^{m_s}$ with $m = m_1 + m_2 + \cdots + m_s$ and

$$\mathcal{K}^{m_i} := \begin{cases} \{(x_1, \tilde{x}^\top)^\top \in \mathbb{R} \times \mathbb{R}^{m_i - 1} \mid x_1 \geq \|\tilde{x}\|\} & (m_i > 1) \\ \mathbb{R}_+ = \{x \in \mathbb{R} \mid x \geq 0\} \quad (m_i = 1). \end{cases}$$

Throughout this paper, $\|\cdot\|$ denotes the Euclidean norm defined by $\|x\| := \sqrt{x^\top x}$, and $\tilde{v}$ denotes $(v_2, v_3, \cdots, v_{m_i-1})^\top \in \mathbb{R}^{m_i-1}$ for $v = (v_1, v_2, \cdots, v_{m_i})^\top \in \mathbb{R}^{m_i}$. For simplicity, we will often write $(x_1, \tilde{x})^\top$ for $(x_1, \tilde{x}^\top)^\top$.

We call the problem (1.1) a semi-infinite second-order cone problem (SISOCP). One of typical applications for SISOCP (1.1) is a Chebychev-like approximation with vector-valued functions. Let $Y \subseteq \mathbb{R}^n$ 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, how can we determine a parameter $u \in \mathbb{R}^\ell$ such that $\Phi(y) \approx F(u, y)$ for all $y \in Y$? One relevant approach is to solve the following problem:

$$\underset{u}{\text{Minimize}} \quad \max_{y\in Y} \; \|\Phi(y) - F(u, y)\| \tag{1.2}$$

which can be rewritten as

$$\begin{array}{ll} \underset{u,r}{\text{Minimize}} & r \\ \text{subject to} & \begin{pmatrix} r \\ \Phi(y) - F(u, y) \end{pmatrix} \in \mathcal{K}^{m+1} \text{ for all } y \in Y \end{array}$$

by introducing the auxiliary variable $r \in \mathbb{R}$. If F is affine with respect to u, then the above problem reduces to SISOCP (1.1) with $\mathcal{K} = \mathcal{K}^{m+1}$.

When $m = 1$ and $\mathcal{K} = \mathbb{R}_+$, SISOCP (1.1) is the classical semi-infinite program (SIP) [3, 5, 8, 13, 15, 16], which has a wide application in engineering (e.g., the air pollution control, the robot trajectory planning, the stress of materials, etc.[8, 13]). So far, many algorithms have been proposed for solving SIPs, such as the discretization method [3], the local reduction based method [4, 11, 18] and the exchange method [5, 6, 16]. The discretization method solves the relaxed SIP with T replaced by a finite set $T^k \subseteq T$, and the sequence of index sets $\{T^k\}$ is generated so that the distance[1] from T^k to T converges to 0 as k goes to infinity. While this method is comprehensible and easy to implement, the computational cost tends to be huge since the cardinality of T^k grows unboundedly. In the local reduction based method, the infinite number of constraints in the SIP is rewritten as a finite number of constraints with implicit functions. Although the SIP can be reformulated as a finitely constrained optimization problem by this method, it is not possible in general to evaluate the implicit functions exactly in a direct manner. The exchange method solves a relaxed subproblem with T replaced by a finite subset $T^k \subseteq T$. In

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

this method, T^k is updated so that $T^{k+1} \subseteq T^k \cup \{t_1, t_2, \cdots, t_r\}$ with $\{t_1, t_2, \cdots, t_r\} \subseteq T \setminus T^k$.

Studies on the second-order cones (SOCs) have been advanced significantly in the last decade. One of the most popular problems associated with SOCs is the linear second-order cone program (LSOCP). The primal-dual interior-point method [1, 12] is well known as an effective algorithm for solving LSOCP, and some software packages implementing them [17, 19] have been produced. The nonlinear second-order cone program (NLSOCP) [9, 10, 20] is more complicated and not studied so much as LSOCP. 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 [2, 7] is one of useful algorithms for solving SOCCP.

The main purpose of the paper is threefold. First, we provide the optimality conditions for SISOCP (1.1). The KKT conditions for SISOCP (1.1) could naturally be described by means of integration and Borel measure since T is infinite. However, we show that, under Slater's constraint qualification, the KKT conditions at the optimum can be represented by using a *finite* number of elements in T. Second, we propose an explicit exchange method for solving the SISOCP (1.1) and show its global convergence under the strict convexity of the objective function. Third, we propose an algorithm that can solve SISOCP (1.1) without the strict convexity. This algorithm is a hybrid of the explicit exchange method and the regularization method, which is known to be effective in handling ill-posed problems. With the help of regularization, a global convergence of the algorithm can be established for SISOCP (1.1) without the strict convexity. As the notation used in this paper, $\mathcal{K} \ni x \perp y \in \mathcal{K}$ denotes the SOC complementarity condition, that is, $x^\top y = 0,\ x \in \mathcal{K}$ and $y \in \mathcal{K}$.

2 Karush-Kuhn-Tucker Conditions

In this section, we show that the KKT conditions for SISOCP (1.1) can be represented with finitely many second-order cone constraints. We first introduce the Slater constraint qualification (SCQ).

Definition 2.1 (SCQ). *We say that the Slater Constraint Qualification* (SCQ) *holds for SISOCP* (1.1) *if there exists a point $x_0 \in \mathbb{R}^n$ such that $A(t)^\top x_0 - b(t) \in \operatorname{int}\mathcal{K}$ for all $t \in T$.*

Under the SCQ, the following theorem holds.

Theorem 2.2 (Theorem 2.12 [14]). *Let $x^* \in \mathbb{R}^n$ be an optimum of SISOCP* (1.1) *and suppose that the SCQ holds for SISOCP* (1.1). *Then, there exist p indices $t_1, t_2, \ldots, t_p \in T$ and Lagrangian multipliers $y^1, y^2, \ldots, y^p \in \mathbb{R}^m$ such that $p \leq n+1$,*

$$\nabla f(x^*) - \sum_{i=1}^{p} A(t_i) y^i = 0,$$
$$\mathcal{K} \ni y^i \perp A(t_i)^\top x^* - b(t_i) \in \mathcal{K} \ \ for \ \ i = 1, 2, \ldots, p.$$

3 Explicit Exchange Method

In this section, we propose an explicit exchange method for solving SISOCP (1.1). Moreover, we show that the algorithm has a global convergence property under mild assumptions. The algorithm proposed in this section requires solving second-order cone programs (SOCP) with finitely many constraints as subproblems. Let SOCP (T') denote the relaxed problem of SISOCP (1.1) with T replaced by a finite subset $T' := \{t_1, t_2, \dots, t_p\} \subseteq T$. Then, the SOCP($T'$) can be formulated as follows:

$$\mathrm{SOCP}(T') \qquad \begin{array}{ll} \text{Minimize} & f(x) \\ \text{subject to} & A(t_j)^\top x - b(t_j) \in \mathcal{K} \;\; (j = 1, 2, \dots, p). \end{array}$$

We suppose that the subproblem SOCP (T') can be solved by any suitable existing algorithm. Let $\bar{x}$ be an optimal solution of SOCP(T'). Then, $\bar{x}$ satisfies the following KKT conditions under some constraint qualification [1, 12]:

$$\begin{aligned} &\nabla f(\bar{x}) - \sum_{t_j \in T'} A(t_j) y_{t_j} = 0, \\ &\mathcal{K} \ni y_{t_j} \perp A(t_j)^\top \bar{x} - b(t_j) \in \mathcal{K} \;\; (j = 1, 2, \dots, p), \end{aligned} \tag{3.1}$$

where y_{t_j} is a Lagrange multiplier vector corresponding to the constraint $A(t_j)^\top \bar{x} - b(t_j) \in \mathcal{K}$ for each j.

Now, we propose the following algorithm.

Algorithm 1 (Explicit exchange method)

Step 0. Choose a positive sequence $\{\gamma_k\} \subseteq \mathbb{R}_{++}$ such that $\lim_{k\to\infty} \gamma_k = 0$. Choose a finite subset $E^0 := \{t_1^0, \dots, t_p^0\} \subseteq T$ and solve SOCP(E^0) to obtain an optimal solution x^0. Set $k := 0$.

Step 1. Set $r := 0$, $T_0 := E^k$ and $v^0 := x^k$. Do the following (a)–(c):

(a) Find a $t_{\mathrm{new}}^r \in T$ such that

$$A(t_{\mathrm{new}}^r)^\top v^r - b(t_{\mathrm{new}}^r) \notin -\gamma_k e + \mathcal{K}. \tag{3.2}$$

If such a t_{new}^r does not exist, i.e.,

$$A(t)^\top v^r - b(t) \in -\gamma_k e + \mathcal{K} \tag{3.3}$$

for any $t \in T$, then set $x^{k+1} := v^r$, $E^{k+1} := T_r$, and go to Step 2. Otherwise, let

$$\overline{T}_{r+1} := T_r \cup \{t_{\mathrm{new}}^r\},$$

and go to (b).

(b) Solve SOCP($\overline{T}_{r+1}$) to obtain an optimum v_{r+1} and Lagrange multipliers y_t^{r+1}, for $t \in \overline{T}_{r+1}$.

(c) Let $T_{r+1} := \{t \in \overline{T}_{r+1} \mid y_t^{r+1} \neq 0\}$. Set $r := r + 1$ and return to (a).

Step 2. If γ_k is sufficiently small, terminate. Otherwise, set $k := k + 1$ and return to Step 1.

In Step 1-(a), $e \in \mathbb{R}^m$ is defined as $e := (e^1, e^2, \dots, e^s)^\top$ and $e^i := (1, 0, \dots, 0)^\top \in \mathcal{K}^{m_i}$. To verify (3.3), we have to solve a certain optimization problem on T and check the nonnegativity of the optimal value. For a detail, see [14]. Since this problem is not necessarily convex, it is not easy to solve it. But, in this paper, we suppose that we can obtain a global optimum of this problem every Step 1. In Step 1-(b), SOCP($\overline{T}_{r+1}$) can be solved by applying an existing method such as the primal-dual interior point method, the regularized smoothing method, and so on [1, 2, 7, 10, 12]. In Step 1-(c), SOCP(T_{r+1}) is obtained from SOCP($\overline{T}_{r+1}$) by removing only the constraints with zero Lagrange multipliers, then the optimal values of those two problems are equal. In addition, the feasible region of SOCP($\overline{T}_{r+1}$) is contained in that of SOCP(T_r). Therefore, we have

$$V_P(T_0) \le V_P(\overline{T}_1) = V_P(T_1) \le \cdots \le V_P(T_r) \le V_P(\overline{T}_{r+1}) = V_P(T_{r+1}) \le \cdots \le V_P(T) < +\infty, \tag{3.4}$$

where $V_P(T')$ denotes the optimal value of SOCP(T').

For the proposed method to be welld-defined, Step 1 have to terminate in finitely many iterations. To ensure this, we suppose the assumptions as follows:

Assumption A. i) Function f is strictly convex over the feasible region of SISOCP (1.1). ii) In Step 1-(b) of Algorithm 1, SOCP($\overline{T}_{r+1}$) is solvable for each r. iii) A sequence generated $\{v^r\}$ in every Step 1 of Algorithm 1 is bounded.

Under the Assumption A, the following theorem holds.

Theorem 3.1. [14, Theorem 4.1] *Let that Assumption A hold. Then, the inner iterations in every Step 1 of Algorithm 1 terminate finitely.*

Moreover, we have the following theorems for the globally convergent property.

Theorem 3.2. [14, Theorem 4.2] *Let Assumption A hold. Let x^* be the optimum*[2] *of SISOCP* (1.1) *, and $\{x^k\}$ be the sequence generated by Algorithm 1. Then, it follows that*

$$\lim_{k\to\infty} x^k = x^*.$$

4 Regularized Explicit Exchange Method

In the previous section, we proposed the explicit exchange method for SISOCP (1.1) and analyzed the convergence property. However, to ensure the global convergence, the strict convexity of the objective function was required (Assumption A). In this section, we propose a regularized explicit exchange method, and establish global convergence of the method without assuming the strict convexity.

[2]From the strictly convexity of f, SISOCP (1.1) has a unique solution.

Let ε be a positive number, and $T' := \{t_1, t_2, \ldots, t_p\}$ be a finite subset of T. The regularized explicit exchange method solves the following SOCP, denoted SOCP(ε, T'), in each iteration.

$$\text{SOCP}(\varepsilon, T') \qquad \begin{array}{ll} \text{Minimize} & \frac{1}{2}\varepsilon\|x\|^2 + f(x) \\ \text{subject to} & A(t_j)^\top x - b(t_j) \in \mathcal{K} \quad (j = 1, 2, \ldots, p). \end{array} \tag{4.1}$$

When the function f is convex, $\frac{1}{2}\varepsilon\|x\|^2 + f(x)$ is strongly convex. Then, if we solve SOCP$(\varepsilon_k, \overline{T}_{r+1})$ with $\varepsilon_k > 0$ instead of SOCP$(\overline{T}_{r+1})$ in Step 1-(b) of Algorithm 1, it is ensured by Theorem 3.1 that the inner iterations terminate[3] finitely. Moreover, by choosing positive sequences $\{\varepsilon_k\}$ and $\{\gamma_k\}$ both converging to 0, the generated sequence is expected to converge to a solution of SISOCP (1.1). Now we propose the following algorithm for SISOCP (1.1).

Algorithm 2 (Regularized Explicit Exchange Method)

Step 0. Choose positive sequences $\{\gamma_k\} \subseteq \mathbb{R}_{++}$ and $\{\varepsilon_k\} \subseteq \mathbb{R}_{++}$ such that $\lim_{k\to\infty} \gamma_k = 0$, $\lim_{k\to\infty} \varepsilon_k = 0$ and $\gamma_k = O(\varepsilon_k)$. Choose a finite subset $E^0 := \{t_1^0, \ldots, t_l^0\} \subseteq T$. Set $k := 0$.

Step 1. Set $r := 0$ and $T_0 := E^k$. Solve SOCP(ε_k, T_0) and let v^0 be an optimum. Do the following (a)–(c):

(a) Find $t_{\text{new}}^r \in T$ such that

$$A(t_{\text{new}}^r)^\top v^r - b(t_{\text{new}}^r) \notin -\gamma_k e + \mathcal{K}. \tag{4.2}$$

If such a t_{new}^r does not exist, i.e.,

$$A(t)^\top v^r - b(t) \in -\gamma_k e + \mathcal{K} \tag{4.3}$$

for any $t \in T$, then set $x^{k+1} := v^r$ and $E^{k+1} := T_r$, and go to Step 2. Otherwise, let

$$\overline{T}_{r+1} := T_r \cup \{t_{\text{new}}^r\},$$

and go to (b).

(b) Solve SOCP$(\varepsilon_k, \overline{T}_{r+1})$ to obtain an optimum v^{r+1} and Lagrange multipliers y_t^{r+1}, for $t \in \overline{T}_{r+1}$.

(c) Let $T_{r+1} := \{t \in \overline{T}_{r+1} \mid y_t^{r+1} \neq 0\}$. Set $r := r + 1$ and return to (a).

Step 2. Both ε_k and γ_k are sufficiently small, then terminate. Otherwise, set $k := k + 1$ and return to Step 1.

---

[3]We can verify Assumption A as follows. Assumption A i) is obvious from the definition of strict and strong convexity. Assumption A ii) holds, since $\operatorname{argmin}\{g(x) \mid x \in X\}$ is nonempty if g is strongly convex and X is closed. Assumption A iii) also holds, since the sequence $\{v^r\}$ generated in Step 1 is contained in the set $L := \{x \mid f(x) \leq V_P(T)\}$ from (3.4), and L is bounded due to the strong convexity of f.

The following theorem shows that a generated sequence globally converges to a solution of SISOCP (1.1).

Theorem 4.1. [14, Lemma 5.1, Thereom 5.2] *Suppose that the SCQ holds for SISOCP (1.1). Let $\{x^k\}$ be a sequence generated by Algorithm 2. Then, $\{x^k\}$ is bounded, and every accumulation point of $\{x^k\}$ solves SISOCP (1.1).*

5 Numerical Experiment

In this section, we report some numerical results with Algorithm 2. The program was coded in Matlab 2008a and run on a machine with an Intel®Core2 Duo E6850 3.00GHz CPU and 4GB RAM. In the experiments, we solve the following SISOCP with a linear objective function:

$$\begin{array}{ll} \text{Minimize} & c^\top x \\ \text{subject to} & A(t)^\top x - b(t) \in \mathcal{K} \quad (\forall t \in T) \end{array} \tag{5.1}$$

with the index set $T = [-1,1]$, $c \in \mathbb{R}^{15}$, $A_{ij}(t) := \alpha_{ij}^0 + \alpha_{ij}^1 t + \alpha_{ij}^2 t^2 + \alpha_{ij}^3 t^3$ $(i = 1,2,\ldots,15,\ j = 1,2,\ldots,30)$ and $b_j(t) := \beta_j^0 + \beta_j^1 t + \beta_j^2 t^2 + \beta_j^3 t^3$ $(j = 1,2,\ldots,30)$. The second-order cone $\mathcal{K} \subseteq \mathbb{R}^{30}$ is chosen to be one of the following four cases: (i) $\mathcal{K} = \mathcal{K}^{30}$, (ii) $\mathcal{K} = \mathcal{K}^{10} \times \mathcal{K}^{20}$, (iii) $\mathcal{K} = (\mathcal{K}^{10})^3 = \mathcal{K}^{10} \times \mathcal{K}^{10} \times \mathcal{K}^{10}$, (iv) $\mathcal{K} = (\mathcal{K}^5)^6 = \mathcal{K}^5 \times \mathcal{K}^5 \times \mathcal{K}^5 \times \mathcal{K}^5 \times \mathcal{K}^5 \times \mathcal{K}^5$. In (5.1), all components of $c \in \mathbb{R}^{15}$ are chosen randomly from $[-2,2]$. β_j^0 $(j = 1,2,\ldots,30)$ are determined so that $(\beta_1^0, \beta_2^0, \ldots, \beta_{30}^0)^\top = e \in \mathbb{R}^{30}$, where e is defined as $e = (e^1, e^2, \ldots, e^s)^\top \in \mathcal{K}^{m_1} \times \cdots \times \mathcal{K}^{m_s}$ and $e^j := (1,0,\ldots,0) \in \mathbb{R}^{m_j}$. In addition, α_{ij}^k, β_i^l $(i = 1,2,\ldots,15,\ j = 1,2,\ldots,30,\ k = 0,1,2,3,\ l = 1,2,3)$ are chosen randomly from $[-2,2]$ so that the origin is an interior feasible point of (5.1). By using $A(t)$ and $b(t)$ generated in this way, (5.1) satisfies the SCQ.

In Step 0 of Algorithm 2, we set $\{\varepsilon_k\}$ and $\{\gamma_k\}$ such that $\varepsilon_k = \gamma_k = 2^{-k}$ for each k. Moreover, we choose 10 points $t_1^0, t_2^0, \ldots, t_{10}^0 \in T$ randomly, and set $E^0 := \{t_1^0, t_2^0, \ldots, t_{10}^0\}$. In Step 1-(a), to find t_{new}^r satisfying (4.2), we first check whether or not (4.2) is satisfied at $t = -1.0$, -0.9, $-0.8, \ldots,$ 0.9, 1.0. If we fail to find t_{new}^r among them, then we solve a certain nonconvex problem (For a detai, see [14]) and check whether or not its optimal value is nonnegative. In Step 1-(b), we solve SOCP$(\varepsilon_k, \overline{T}_{r+1})$ by the smoothing method [2, 7]. In Step 2, we terminate the algorithm if $\max(\varepsilon_k, \gamma_k) \leq 10^{-6}$, which means that we always stop the algorithm when $k = 20$ since $\varepsilon_{20} = \gamma_{20} = 2^{-20} < 10^{-6}$.

The obtained results are shown in Table 1, Table 2, Table 3 and Table 4, in which cpu(s), t_{add} and E_{fin} denote the CPU time in seconds, the cumulative number of times t_{new}^r is added to T_r in Step 1, and the value of E^k at the termination of the algorithm, respectively.

From the tables, we can observe that the computational cost tends to be higher as the number of SOCs in $\mathcal{K}$ gets larger. For example, in the case of $\mathcal{K} = \mathcal{K}^{30}$ (one SOC in $\mathcal{K}$), the cpu time is only 5 seconds or so, whereas it becomes around 20 seconds when $\mathcal{K} = (\mathcal{K}^5)^6$ (six SOCs in $\mathcal{K}$). We also note that both $t = -1$ and 1 (the extreme points of T) belong to E_{fin} for all the test problems. This implies that the constraints $A(t)^\top x - b(t) \in \mathcal{K}$ with $t = -1$ and 1 are both active at the optimal solutions. For the problems with $E_{\rm fin} = \{-1, 1\}$, the values of cpu(s) and $t_{\rm add}$ seem relatively small. In fact, for such problems, the active sets at the optima can often be identified in a small number of iterations. On the other hand, if $E_{\rm fin}$ has elements other than -1 or 1, then the values of cpu(s) and $t_{\rm add}$ tend to be larger. Especially, problems A4, B3, C2 and D9 yield the largest values among the test problems with $\mathcal{K} = \mathcal{K}^{30}$, $\mathcal{K} = \mathcal{K}^{10} \times \mathcal{K}^{20}$, $\mathcal{K} = (\mathcal{K}^{10})^3$, and $\mathcal{K} = (\mathcal{K}^5)^6$, respectively. Indeed, for those four problems, $E_{\rm fin}$ has the *third* element that could not be identified at an early stage of the iterations.

Table 1: Results for $\mathcal{K} = \mathcal{K}^{30}$

problem	cpu(s)	$t_{\rm add}$	$E_{\rm fin}$
A1	4.78	3	$\{-1, 1\}$
A2	5.27	3	$\{-1, 1\}$
A3	5.70	3	$\{-1, 1\}$
A4	8.39	7	$\{-1, 0.14, 1\}$
A5	3.94	3	$\{-1, 1\}$
A6	4.51	6	$\{-1, 1\}$
A7	4.99	3	$\{-1, 1\}$
A8	4.70	3	$\{-1, 1\}$
A9	4.52	4	$\{-1, 1\}$
A10	4.75	3	$\{-1, 1\}$

Table 2: Results for $\mathcal{K} = \mathcal{K}^{10} \times \mathcal{K}^{20}$

problem	cpu(s)	$t_{\rm add}$	$E_{\rm fin}$
B1	9.79	7	$\{-1, 1\}$
B2	6.31	6	$\{-1, 1\}$
B3	10.02	15	$\{-1, 0.3, 1\}$
B4	6.90	3	$\{-1, -0.24, 1\}$
B5	6.63	6	$\{-1, 0.4, 1\}$
B6	8.23	8	$\{-1, 1\}$
B7	6.54	4	$\{-1, -0.14, 1\}$
B8	7.32	6	$\{-1, 1\}$
B9	6.78	2	$\{-1, 1\}$
B10	6.29	5	$\{-1, 1\}$

Table 3: Results for $\mathcal{K} = (\mathcal{K}^{10})^3$

problem	cpu(s)	t_{add}	E_{fin}
C1	7.85	2	$\{-1, 1\}$
C2	15.69	7	$\{-1, 0.16, 1\}$
C3	9.72	7	$\{-1, -0.38, 1\}$
C4	10.16	3	$\{-1, 1\}$
C5	7.31	2	$\{-1, 1\}$
C6	7.76	5	$\{-1, 1\}$
C7	7.73	5	$\{-1, 1\}$
C8	8.97	4	$\{-1, 1\}$
C9	10.78	5	$\{-1, 1\}$
C10	7.23	2	$\{-1, 1\}$

Table 4: Results for $\mathcal{K} = (\mathcal{K}^{5})^6$

problem	cpu(s)	t_{add}	E_{fin}
D1	27.00	7	$\{-1, -0.58, 1\}$
D2	20.10	14	$\{-1, 0.36, 1\}$
D3	17.42	6	$\{-1, 0.90, 1\}$
D4	26.16	6	$\{-1, -0.56, 1\}$
D5	20.52	6	$\{-1, 0.07, 1\}$
D6	27.57	13	$\{-1, 0.33, 1\}$
D7	25.96	8	$\{-1, -0.73, 1\}$
D8	22.64	4	$\{-1, 1\}$
D9	65.15	26	$\{-1, -0.63, -0.41, -0.23, 0.25, 1\}$
D10	20.12	8	$\{-1, 1\}$

References

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

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

[3] M. A. Goberna and M. A. López. *Semi-infinite Programming: Recent Advances.* Kluwer Academic Publishers, Dordrecht, 2001.

[4] G. Gramlich, R. Hettich, and E. W. Sachs. Local convergence of SQP methods in semi-infinite programming. *SIAM Journal on Optimization*, 5:641–658, 1995.

[5] H. C. Lai and S.-Y. Wu . On linear semi-infinite programming problems. *Numerical Functional Analysis and Optimization*, 13:287–304, 1992.

[6] 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*, 175:335–353, 2005.

[7] S. Hayashi, N. Yamashita, and M. Fukushima. A combined smoothing and regularization method for monotone second-order cone complementarity problems. *SIAM Journal on Optimization*, 15:593–615, 2005.

[8] R. Hettich and K. O. Kortanek. Semi-infinite programming: Theory, methods, and applications. *SIAM Review*, 35:380–429, 1993.

[9] C. Kanzow, I. Ferenczi, and M. Fukushima. On the local convergence of semismooth Newton methods for linear and nonlinear second-order cone programs without strict complementarity. *SIAM Journal on Optimization*, 20:297–320, 2009.

[10] H. Kato and M. Fukushima. An SQP-type algorithm for nonlinear second-order cone programs. *Optimization Letters*, 1:129–144, 2007.

[11] C. Ling, L. Qi, G. L. Zhou, and S.-Y. Wu. Global convergence of a robust smoothing SQP method for semi-infinite programming. *Journal of Optimization Theory and Applications*, 129:147–164, 2006.

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

[13] M. López and G. Still. Semi-infinite programming. *European Journal of Operational Research*, 180:491–518, 2007.

[14] T. Okuno. A regularized explicit exchange method for semi-infinite programs with an infinite number of second-order cone constraints. Master's thesis, Department of Applied Mathematics and Physics, Graduate School of Informatics, Kyoto University, 2009.

[15] R. Reemtsen and J. Rückmann. *Semi-infinite programming.* Kluwer Academic Publishers, Boston, 1998.

[16] S.-Y. Wu and D. H. Li and L. Qi and G. Zhou. An iterative method for solving KKT system of the semi-infinte programming. *Optimization Methods and Software*, 20:629–643, 2005.

[17] J. F. Sturm. Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. *Optimization Methods and Software*, 11:625–653, 1999.

[18] Y. Tanaka, M. Fukushima, and T. Ibaraki. A globally convergent SQP method for semi-infinite nonlinear optimization. *Journal of Computational and Applied Mathematics*, 23:141–153, 1988.

[19] K. C. Toh, M. J. Todd, and R. H. Tütüncü. SDPT3—a MATLAB software package for semidefinite programming, version 2.1. *Optimization Methods and Software*, 11:545–581, 1999.

[20] H. Yamashita and H. Yabe. A primal-dual interior point method for nonlinear optimization over second-order cones. *Optimization Methods and Software*, 24:407–426, 2009.