Master's Thesis

An exchange method with refined subproblems for convex semi-infinite programming problems

Guidance

Associate Professor Nobuo YAMASHITA

Kensuke GOMOTO

Department of Applied Mathematics and Physics

Graduate School of Informatics

Kyoto University

February 2014

Abstract

The semi-infinite programming problem (SIP) is an optimization problem with an infinite number of constraints in a finite dimensional space. The SIP has been studied extensively so far, since a lot of practical problems in various fields such as physics, economics, and engineering can be formulated as the SIPs. The exchange method is one of the most useful algorithms for solving the SIP, and it has been developed by many researchers. In this paper, we focus on the convex SIPs and propose a new exchange method for solving them. While the traditional exchange method solves a sequence of the relaxed problems with finitely many constraints that are selected from the original constraints, our method solves a sequence of semi-infinite programs relaxing the original SIP. These relaxed problems can be solved efficiently by transforming them into certain optimization problems with finitely many constraints. Moreover, under some mild assumptions, they approximate the original SIP more precisely than the finite relaxed problems in the traditional exchange method. We also establish global convergence of the proposed method under strict convexity assumption on the objective function, and examine its efficiency through some numerical experiences.

Contents

1 Introduction

The semi-infinite programming problem (SIP) is an optimization problem with a finite dimensional variable $x \in \mathbb{R}^n$ and an infinite number of inequality constraints. The SIP has been studied extensively so far since there are a lot of applications such as Chebyshev approximation in mathematics, optimal control and trajectory control in engineerings, air/water pollution control problem, and production planning, etc. Also, from the 1960s, there have been many theoretical studies such as the optimality condition and duality theorem [9].

In this paper, we focus on the following convex SIP:

$$\begin{aligned} \text{SIP :} \qquad & \underset{x \in X}{\text{minimize}} && f(x) \\ & \text{subject to} && g(x,t) \leq 0 \quad \forall t \in T, \end{aligned} \tag{1.1}$$

where $X \subseteq \mathbb{R}^n$ is a compact convex set, $T \subseteq \mathbb{R}^m$ is a nonempty compact set of the form $T = \{t \in \mathbb{R}^m | At \leq b\}$ with $A \in \mathbb{R}^{l \times m}$ and $b \in \mathbb{R}^l$, $f : \mathbb{R}^n \to \mathbb{R}$ is a function differentiable and convex over X, and $g : \mathbb{R}^n \times T \to \mathbb{R}$ is a function differentiable for any $(x,t) \in X \times T$ and convex with respect to x. Since t plays a role of index in a finitely constrained optimization problem, t and T are called index and index set, respectively.

Many algorithms for solving SIP have been studied so far. Among them, the discretization method and the exchange methods are well known methods, both of which require to solve finitely approximated subproblems in each iteration. The discretization method generates a sequence of index sets $\{T_k\} \subseteq T$ satisfying $|T_k| < \infty$, $T_0 \subset T_1 \subset T_2 \subset \cdots \subset T$ and $\lim_{k\to\infty} \operatorname{dist}(T_k, T) = 0$[1]. Then, in each iteration k, it solves the finitely approximated subproblem with respect to T_k to obtain the optimum x^k, so that x^k converges to the original SIP optimum as k goes infinity [7, 10, 11]. On the other hand, the exchange method generates the sequence converging to the SIP optimum by exchanging an index belonging to T_k by another index belonging to $T \setminus T_k$ [1, 4, 5, 6]. Unlike the discretization method, the computational cost for each subproblem does not become very large, since $|T_k|$ is bounded even when $k \to \infty$.

In this paper, we propose an exchange algorithm in which each subproblem is generated by means of the first order approximation with respect to t. Although the subproblems are still SIPs, they can be transformed into the problems with a finite number of constraints equivalently. Moreover, if $\nabla_t g$ is Lipschitzian, then each subproblem approximates the original SIP more precisely than the existing exchange methods. Consequently, we can expect that our method finds the optimal solution in a lower number of iterations.

[1]For two sets S and T with $S \subset T$, the distance from S to T is defined as $\operatorname{dist}(S,T) = \sup_{t\in T} \inf_{s \in S} \|s - t\|$.

This paper is organized as follows. In Section 2, we give some mathematical preliminaries that will be useful in the subsequent analyses. In Section 3, we propose an algorithm and mention some properties. In Section 4, we show the global convergence of the algorithm under some assumptions. In Section 5, we provide some techniques how to solve each subproblem and how to treat the constant necessary for the numerical experiments. In Section 6, we give some numerical results relevant to Chebyshev approximation problem. Finally in Section 7, we conclude the paper with some remarks.

2 Preliminaries

In this section, we give some preliminaries that will be useful in the subsequent sections. We first define the closedness, properness and convexity of functions.

Definition 2.1 *For a given function* $f:\ \mathbb{R}^n \to (-\infty, \infty]$*, we denote the effective domain of* f *by* $\operatorname{dom} f := \{x \in \mathbb{R}^n \mid f(x) < \infty\}$*. Then we say that*

i. the function f *is proper if* $\operatorname{dom} f \neq \emptyset$;

ii. the function f *is closed if* f *is lower semicontinuous*;

iii. the function f *is convex if*

$$f(\theta x + (1-\theta)y) \leq \theta f(x) + (1-\theta) f(y)$$

for any $\theta \in [0,1]$ *and* $(x,y) \in \mathbb{R}^n \times \mathbb{R}^n$*.*

Then, we have the following proposition on the level set.

Proposition 2.1 [8, Corollary 8.7.1] *Let* f *be a proper closed convex function. Suppose that there exists an* $\alpha \in \mathbb{R}$ *such that the level set* $\{x \in \mathbb{R}^n \mid f(x) \leq \alpha\}$ *is compact. Then* $\{x \in \mathbb{R}^n \mid f(x) \leq \beta\}$ *is compact for any* $\beta \in \mathbb{R}$ *if it is nonempty.*

Next, we give the following two propositions which concern the convexity and differentiability of a function defined as a maximum of finitely or infinitely many functions.

Proposition 2.2 *Let* $T \subseteq \mathbb{R}^m$ *be a nonempty compact set, and* $g : \mathbb{R}^n \times T \to \mathbb{R}$ *be a function such that* $g(\cdot, t)$ *is convex for any fixed* $t \in T$*. Then, the function* $\overline{g}$ *defined by* $\overline{g}(x) := \sup_{t \in T} g(x,t)$ *is convex.*

Proof It suffices to show that

$$\overline{g}(\theta x + (1-\theta)y) \leq \theta \overline{g}(x) + (1-\theta)\overline{g}(y)$$

for any $\theta \in [0,1]$ and $(x,y) \in \mathbb{R}^n \times \mathbb{R}^n$. Then we have

$$\begin{aligned}
& \theta\overline{g}(x) + (1-\theta)\overline{g}(y) - \overline{g}(\theta x + (1-\theta)y) \\
= \;& \theta \sup_{t\in T} g(x,t) + (1-\theta)\sup_{t\in T} g(y,t) - \sup_{t\in T} g(\theta x + (1-\theta)y, t) \\
\geq \;& \sup_{t\in T}\{\theta g(x,t) + (1-\theta) g(y,t)\} - \sup_{t\in T} g(\theta x + (1-\theta)y, t) \\
\geq \;& 0,
\end{aligned}$$

where the last inequality follows from the convexity of $g(\cdot, t)$. ■

Proposition 2.3 [3, Theorem 10.2.1] *Let $T \subseteq \mathbb{R}^m$ be a nonempty compact set, and $d : \mathbb{R}^n \times T \to \mathbb{R}$ be a function such that $d(\cdot, t)$ is continuously differentiable for any fixed $t \in T$. Suppose that* $\operatorname{argmax}_{t\in T} d(x,t)$ *is a singleton, say $\bar{t}(x)$, for any $x \in \mathbb{R}^n$. Then, the function $\overline{d}$ defined by $\overline{d}(x) := \max_{t\in T} d(x,t)$ is differentiable and*

$$\nabla \overline{d}(x) = \nabla_x d(x, \bar{t}(x)).$$

The next proposition shows that the property of a function whose gradient is Lipschitz continuous.

Proposition 2.4 *Let $h : \mathbb{R}^m \to \mathbb{R}$ be a continuously differentiable function such that ∇h is Lipschitz continuous. Then, we have*

$$h(t_1) + \nabla h(t_1)^\top t_2 - \frac{L}{2}\|t_2\|^2 \leq h(t_1 + t_2), \;\; \forall t_1, t_2 \in \mathbb{R}^m$$

for any $(t_1, t_2) \in \mathbb{R}^m \times \mathbb{R}^m$, where L is the Lipschitz constant for ∇h.

Proof Fix t_1 and t_2 arbitrarily, and let $p(s) = h(t_1 + st_2)$ for $s \in \mathbb{R}$. Then we have $\frac{dp}{ds}(s) = \nabla h(t_1 + st_2)^\top t_2$. Thus,

$$\begin{aligned}
h(t_1) + \nabla h(t_1)^\top t_2 - h(t_1 + t_2) &= p(0) + \nabla h(t_1)^\top t_2 - p(1) \\
&= -\int_0^1 \frac{dp}{ds}(s)ds + \nabla h(t_1)^\top t_2 \\
&= -\int_0^1 \nabla h(t_1 + st_2)^\top t_2 ds + \nabla h(t_1)^\top t_2 \\
&= -\int_0^1 (\nabla h(t_1 + st_2) - \nabla h(t_1))^\top t_2 ds \\
&\leq \int_0^1 \|\nabla h(t_1 + st_2) - \nabla h(t_1)\| \|t_2\| ds \\
&\leq \int_0^1 Ls\|t_2\|^2 ds = L\|t_2\|^2 \int_0^1 s ds = \frac{1}{2}L\|t_2\|^2.
\end{aligned}$$

This completes the proof. ■

3 Algorithm

In this section, we propose a new exchange algorithm. Let

$$E := \{\bar{t}^1, \bar{t}^2, \ldots, \bar{t}^p\} \tag{3.1}$$

be a finite subset of T. In the existing exchange method, each subproblem was of the form

$$\begin{array}{ll} \underset{x \in X}{\text{minimize}} & f(x) \\ \text{subject to} & g(x, \bar{t}^1) \le 0, \\ & \vdots \\ & g(x, \bar{t}^p) \le 0. \end{array} \tag{3.2}$$

On the other hand, we solve the following subproblem at each iteration:

$$\begin{array}{lll} & \underset{x \in X}{\text{minimize}} & f(x) \\ \text{NLP}(E) & \text{subject to} & g(x, \bar{t}^1) + \nabla_t g(x, \bar{t}^1)^\top (t - \bar{t}^1) - \dfrac{L}{2}\|t - \bar{t}^1\|^2 \le 0 \ (\forall t \in T), \\ & & \vdots \\ & & g(x, \bar{t}^p) + \nabla_t g(x, \bar{t}^p)^\top (t - \bar{t}^p) - \dfrac{L}{2}\|t - \bar{t}^p\|^2 \le 0 \ (\forall t \in T), \end{array}$$

where $L \in \mathbb{R}$ is the Lipschitz constant of $\nabla_t g(x, \cdot)$ for an arbitrarily fixed x, i.e.,

$$\|\nabla_t g(x, t_1) - \nabla_t g(x, t_2)\| \le L\|t_1 - t_2\| \tag{3.3}$$

for any $t_1, t_2 \in T \times T$ and $x \in X$. Then, for

$$\tilde{F}(E) := \{\text{feasible region of NLP}(E)\}, \tag{3.4}$$

$$F(E) := \{\text{feasible region of problem (3.2)}\}, \tag{3.5}$$

$$F(T) := \{\text{feasible region of SIP(1.1)}\}, \tag{3.6}$$

we have the following proposition.

Proposition 3.1 *Let $E \subseteq T$ be an arbitrary finite subset as in* (3.1). *Let $\tilde{F}(E)$, $F(E)$, and $F(T)$ be defined as* (3.4)-(3.6), *respectively. Then, we have*

$$F(T) \subseteq \tilde{F}(E) \subseteq F(E).$$

Proof From Proposition 2.4, we have

$$g(x, \bar{t}^i) + \nabla_t g(x, \bar{t}^i)^\top (t - \bar{t}^i) - \frac{L}{2}\|t - \bar{t}^i\|^2 \le g(x, t) \tag{3.7}$$

for each $i = 1, \dots, p$ and $t \in T$. Hence, we have $F(T) \subseteq \tilde{F}(E)$. On the other hand, we have $\tilde{F}(E) \subseteq F(E)$ since

$$g(x, \bar{t}^i) = g(x, \bar{t}^i) + \nabla_t g(x, \bar{t}^i)^\top (\bar{t}^i - \bar{t}^i) - \frac{L}{2}\|\bar{t}^i - \bar{t}^i\|^2 \le 0,$$

for any $x \in \tilde{F}(E)$ and $\bar{t}^i \in E$, where the inequality is due to $\bar{t}^i \in E \subseteq T$. This completes the proof. ∎

This proposition implies that NLP(E) approximates the original SIP more precisely than existing exchange methods, and therefore we can expect that our method finds the original SIP optimum more rapidly than existing exchange methods. One may think that NLP(E) is as difficult as (1.1) since NLP(E) still has an infinite number of inequality constraints. However, NLP(E) can be transformed into an optimization problem with a finite number of constraints equivalently by using the duality theory for the quadratic programs. We provide the transformation techniques in the subsequent section.

The details of the algorithm are as follows.

Algorithm 3.1

Step 0: *Set $k := 0$. Choose a finite subset $T^0 \subset T$. Let $\{\gamma_k\}$ be a positive sequence such that $\lim_{k\to\infty} \gamma_k = 0$, and L be the Lipschitz constant satisfying* (3.3).

Step 1: *Obtain x^k and T^k by the following steps.*

- **Step 1-0:** *Set $r := 0, E^0 := T^k$, and solve* NLP(E^0) *to obtain the optimum v^0.*
- **Step 1-1:** *Find a t^r_{new} such that $g(v^r, t^r_{\text{new}}) > \gamma_k$. If such a t^r_{new} does not exist, then let $x^{k+1} = v^r, T^{k+1} = E^r$, and go to Step* 2. *Otherwise, let $\overline{E}^{r+1} = E^r \cup \{P_T(t + \frac{1}{L}\nabla_t g(v^r, t)) \,|\, t \in E^r\} \cup \{t^r_{\text{new}}\}$ and go to Step* 1-2.
- **Step 1-2:** *Solve* NLP($\overline{E}^{r+1}$) *to obtain its optimum v^{r+1} and the corresponding Lagrange multiplier λ^{r+1}.*
- **Step 1-3:** *Let $E^{r+1} := \{\bar{t} \in \overline{E}^{r+1} \,|\, \lambda^{r+1}_{\bar{t}} \neq 0\}$. Set $r := r+1$ and return to Step* 1-1.

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

In Step 1-1, $P_T(\cdot)$ denotes the projection onto the index set T, i.e.,

$$P_T(s) := \operatorname*{argmin}_{t\in T} \|s - t\|.$$

Notice that we add not only t^r_{new} but also $\{P_T(t + \frac{1}{L}\nabla_t g(v^r, t)) \,|\, t \in E^r\}$ to E^r. This is because $P_T(t+\frac{1}{L}\nabla_t g(v^r, t))$ is more desirable than t in the sense that $P_T(t+\frac{1}{L}\nabla_t g(v^r, t))$

is obtained by means of the steepest ascent method with respect to t for a fixed v^r. In Step 1-3, $\lambda_{\bar{t}}^{r+1}$ denotes the Lagrange multiplier corresponding to the constraint of the index $\bar{t}$. Here, we remove inactive indices whose Lagrange multipliers are zero. In the subsequent convergence analysis, we omit the termination condition in Step 2, so that the algorithm may generate an infinite sequence.

4 Global convergence

In this section, we show that the sequence generated by Algorithm 3.1 converges to the SIP optimum. To this end, we make the following assumption.

Assumption 4.1 **(i)** *Function f is strictly convex.*

(ii) *For an arbitrarily fixed $t \in T$, $g(\cdot, t)$ is Lipschitz continuous, that is, there exists a constant $M > 0$ such that*

$$\|g(x,t) - g(y,t)\| \leq M\|x - y\|$$

for any $(x, y) \in \mathbb{R}^n \times \mathbb{R}^n$.

(iii) *For an arbitrarily fixed $x \in X$, $\nabla_t g(x, \cdot)$ is Lipschitz continuous, that is, there exits a constant $L > 0$ such that*

$$\|\nabla_t g(x, t_1) - \nabla_t g(x, t_2)\| \leq L\|t_1 - t_2\|$$

for any $(t_1, t_2) \in T \times T$.

(iv) *Function*

$$g_{\bar{t}}(x) := \max_{t \in T} \left\{ g(x, \bar{t}) + \nabla_t g(x, \bar{t})^\top (t - \bar{t}) - \frac{L}{2}\|t - \bar{t}\|^2 \right\} \tag{4.1}$$

is convex for any $\bar{t} \in E_r$.

(v) *$\{v^r\}$ is bounded for each k.*

Notice that assumptions (ii) and (iv) hold when g is affine with respect to x, and (v) holds when f is strongly convex. Moreover, by using the function $g_{\bar{t}}$ defined by (4.1), NLP(E) can be rewritten equivalently as

$$\begin{array}{lll} & \underset{x \in X}{\text{minimize}} & f(x) \\ \text{NLP}(E) & \text{subject to} & g_{\bar{t}^1}(x) \leq 0, \\ & & \vdots \\ & & g_{\bar{t}^p}(x) \leq 0, \end{array}$$

when $E \in T$ is given by (3.1).

Under Assumptions 4.1, we show the global convergence of Algorithm 3.1. First, we provide the following proposition stating the differentiability of function $g_{\bar{t}}$.

Proposition 4.1 *For any $\bar{t} \in E_r$, the function $g_{\bar{t}}$ defined by* (4.1) *is differentiable.*

Proof Since $\hat{g}(x,t) := g(x,\bar{t}) + \nabla_t g(x,\bar{t})^\top (t-\bar{t}) - \frac{L}{2}\|t-\bar{t}\|^2$ is strongly concave with respect to t, we have that $\operatorname{argmax}_{t\in T} \hat{g}(x,t)$ is a singleton. Hence, by Proposition 2.3 $\hat{g}(x,t)$ is differentiable with respect to x. ■

The following proposition states that the distance between v^r and v^{r+1} does not tend to zero during the inner iterations in Step1.

Proposition 4.2 *Suppose that Assumption* 4.1 *holds. Then, there exists an $M > 0$ such that*

$$\|v^{r+1} - v^r\| \geq \frac{\gamma_k}{M} \tag{4.2}$$

for any $r \geq 0$ and $k \geq 0$.

Proof Fix k arbitrarily. From Assumption 4.1(ii), there exists a positive number $M_k > 0$ such that

$$\|g(v^{r+1},t) - g(v^r,t)\| \leq M_k\|v^{r+1} - v^r\| \tag{4.3}$$

for any $t \in T$ and r. Moreover, since it follows $g(v^r, t^r_{\text{new}}) > \gamma_k$ and $g(v^{r+1}, t^r_{\text{new}}) \leq 0$ from Steps 1-1 and 1-2, respectively, we have

$$\|g(v^{r+1}, t^r_{\text{new}}) - g(v^r, t^r_{\text{new}})\| \geq \gamma_k. \tag{4.4}$$

From (4.3) and (4.4),

$$\gamma_k \leq \|g(v^{r+1}, t^r_{\text{new}}) - g(v^r, t^r_{\text{new}})\| \leq M_k\|v^{r+1} - v^r\|,$$

that is,

$$\|v^{r+1} - v^r\| \geq \frac{\gamma_k}{M_k}.$$

Letting $M := \max_k M_k$, we have (4.2). ■

Next, we show that the inner iterations of Step 1 terminate finitely.

Theorem 4.1 *Suppose that Assumption* 4.1 *holds. Then, the inner iteration in Step* 1 *of Algorithm* 4.1 *terminates finitely for each k.*

Proof Suppose, for contradiction, that the inner iteration does not terminate finitely for some k. Since $\{v^r\}$ is bounded by Assumption 4.1 (v), there exist accumulation points v^* and v^{**} of $\{v^r\}$ such that $v^{r_j} \to v^*$ and $v^{r_{j+1}} \to v^{**}$ as $j \to \infty$. Moreover, we must have $v^* \neq v^{**}$ from Proposition 4.2. Since $g_{\bar{t}}$ is differentiable by Proposition 4.1 and v^r solves NLP($\overline{E}^r$), we have the following KKT conditions:

$$
\begin{aligned}
\nabla f(v^r) + \sum_{\bar{t} \in \overline{E}^r} \lambda_{\bar{t}}^r \nabla g_{\bar{t}}(v^r) &= 0, \\
\lambda_{\bar{t}}^r \geq 0, g_{\bar{t}}(v^r) \leq 0, \lambda_{\bar{t}}^r g_{\bar{t}}(v^r) &= 0 \ (\bar{t} \in \overline{E}^r),
\end{aligned} \tag{4.5}
$$

where $\lambda_{\bar{t}}^r$ is the Lagrange multiplier. Now, let us denote the optimal values of (1.1) and NLP(E) by $V(T)$ and $V(E)$, respectively. Then, from Step 1-3, we have $V(E^{r+1}) = V(\overline{E}^{r+1})$. Also we have $V(\overline{E}^{r+1}) \geq V(E^r)$ since $\overline{E}^{r+1} \subseteq E^r$. Consequently, we have $V(E^0) \leq V(\overline{E}^1) = V(E^1) \leq \cdots \leq V(E^r) \leq V(\overline{E}^{r+1}) = V(E^{r+1}) \leq \ldots \leq V(T) < \infty$, i.e.,

$$f(v^1) \leq f(v^2) \leq \ldots \leq V(T) < \infty,$$

Let $F_r := f(v^{r+1}) - f(v^r) - \nabla f(v^r)^\top (v^{r+1} - v^r)$. Then, we have

$$
\begin{aligned}
f(v^{r+1}) - f(v^r) &= F_r - \nabla f(v^r)^\top (v^{r+1} - v^r) \\
&= F_r - \left(\sum_{\bar{t} \in \overline{E}^r} \lambda_{\bar{t}}^r \nabla g_{\bar{t}}(x) \right)^\top (v^{r+1} - v^r) \\
&\geq F_r - \sum_{\bar{t} \in \overline{E}^r} \left\{ \lambda_{\bar{t}}^r (g_{\bar{t}}(v^{r+1}) - g_{\bar{t}}(v^r)) \right\} \\
&= F_r - \sum_{\bar{t} \in \overline{E}^r} \lambda_{\bar{t}}^r g_{\bar{t}}(v^{r+1}) \\
&= F_r - \sum_{\bar{t} \in E^r} \lambda_{\bar{t}}^r g_{\bar{t}}(v^{r+1}) - \sum_{\bar{t} \in \overline{E}^r \setminus E^r} \lambda_{\bar{t}}^r g_{\bar{t}}(v^{r+1}) \\
&\geq F_r \geq 0.
\end{aligned}
$$

where the first inequality follows from Assumption 4.1 (iv) and the third equality follows from (4.5). In addition, the second inequality holds since $E^r \subseteq \overline{E}^{r+1}$ and $\lambda_{\bar{t}}^r = 0$ for any $\bar{t} \in \overline{E}^r \setminus E^r$.

Therefore, we have

$$0 = \lim_{r \to \infty} F_r = \lim_{j \to \infty} F_{r_j} = f(v^{**}) - f(v^*) - \nabla f(v^*)^\top (v^{**} - v^*).$$

However, this contradicts the strictly convexity of f since $v^* \neq v^{**}$. Hence, the inner iterations of Step 1 must terminate finitely for each k. ∎

The next theorem shows the global convergence of Algorithm 3.1 .

Theorem 4.2 *Suppose that Assumption* 4.1 *holds. Let* x^* *be the optimum of* SIP(1.1), *and* $\{x^k\}$ *be the sequence generated by Algorithm* 3.1. *Then, we have*

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

Proof First, we show that $\{x^k\}$ is bounded. Let $X(\gamma) := \{x \in \mathbb{R}^n \mid g(x,t) \le \gamma, \forall t \in T\}, \Lambda := \{x \in \mathbb{R}^n \mid f(x) \le f(x^*)\}$, and $\bar{\gamma} := \max_{k\ge 0} \gamma_k$. Since $x^{k+1} \in \Lambda \cap X(\gamma_k) \subseteq \Lambda \cap X(\bar{\gamma})$, it suffices to show that $\Lambda \cap X(\gamma)$ is bounded for any γ. Let $h(x) := \max_{t\in T} g(x,t)$. Then,

$$X(\gamma) = \{x \in \mathbb{R}^n \mid h(x) \le \gamma\}.$$

Since T is compact and $g(x,t)$ is continuous, $h(x) < \infty$. Moreover, h is a proper closed convex function since $g(\cdot,t)$ is convex and continuous from Proposition 2.2. Let $\bar{h} : \mathbb{R}^n \to (-\infty, +\infty]$ be defined as

$$\bar{h}(x) := \begin{cases} h(x) & (x \in \Lambda) \\ \infty & (x \notin \Lambda). \end{cases}$$

Then, $\bar{h}$ is also a proper closed convex function since Λ is convex. Thus we have

$$\Lambda \cap X(\gamma) = \{x \in \mathbb{R}^n \mid \bar{h}(x) \le \gamma\},$$

i.e., $\Lambda \cap X(\gamma)$ is a level set of function $\bar{h}$. Notice that $L \cap X(0) = \{x^*\}$ since f is strictly convex. Hence, by Proposition 2.1, $L \cap X(\gamma)$ is compact for any γ.

Next, we show that $\lim_{k\to\infty} x^k = x^*$. Let $\overline{x}$ be any accumulation point of $\{x^k\}$. Then, there exists a subsequence $\{x^{k_j}\} \subseteq \{x^k\}$ such that $\lim_{j\to\infty} x^{k_j} = \overline{x}$. For any j and $t \in T$, we have $g(x^{k_j}, t) \le \gamma_{k_j}$ and $f(x^{k_j}) \le f(x^*)$. Thus, by letting $j \to \infty$, we have

$$g(\overline{x}, t) \le 0 \ (\forall t \in T), \tag{4.6}$$

$$f(\overline{x}) \le f(x^*), \tag{4.7}$$

from the continuity of f and g. Since (4.6) implies the feasibility of $\overline{x}$, we must have $f(\overline{x}) \ge f(x^*)$, which together with (4.7) implies $f(\overline{x}) = f(x^*)$. Therefore, $\overline{x}$ also solves (1.1). Since f is strictly convex, we must have $\overline{x} = x^*$. We thus have $\lim_{k\to\infty} x^k = x^*$. ■

5 Implementation issues

In this section, we state some technical issues on implementing the proposed algorithm. Especially, we provide some techniques for solving NLP(E^0) and NLP($\overline{E}^{r+1}$) in Step 1, and how to choose L in Step 0 appropriately.

5.1 How to solve the subproblems

In this section, we show how to solve the subproblems NLP(E^0), and NLP($\overline{E}^{r+1}$) in Step 1. We note that it depends on whether or not the projection $P_T(\cdot)$ in Step 1-1 can be calculated explicitly.

We first consider the case where we have the explicit expression of P_T. Let $E \subseteq T$ be an arbitrary finite set given as in (3.1). Then, for each $i = 1, 2, \ldots, p$, we have

$$\operatorname*{argmax}_{t\in T} \left\{ g(x, \bar{t}^i) + \nabla_t g(x, \bar{t}^i)^\top (t - \bar{t}^i) - \frac{L}{2}\|t - \bar{t}^i\|^2 \right\} = P_T\left(\bar{t}^i + \frac{1}{L}\nabla_t g(x, \bar{t}^i) \right).$$

Thus NLP(E) can be cast as the following optimization problem with a finite number of constraints:

$$\begin{array}{ll} \underset{x\in X}{\text{minimize}} & f(x) \\ \text{subject to} & g(x, \bar{t}^i) + \nabla_t g(x, \bar{t}^i)^\top \left(P_T\left(\bar{t}^i + \frac{1}{L}\nabla_t g(x, \bar{t}^i) \right) - \bar{t}^i \right) \\ & \quad - \frac{L}{2}\left\| P_T\left(\bar{t}^i + \frac{1}{L}\nabla_t g(x, \bar{t}^i) \right) - \bar{t}^i \right\|^2 \le 0 \ (i = 1, 2, \ldots, p). \end{array} \tag{5.1}$$

For example, when T is represented by means of box constraints, the projection P_T is represented explicitly, and the above problem can be solved by an existing algorithm.

We next consider the case where P_T cannot be calculated explicitly. Let $E \subseteq T$ be an arbitrary finite set given as in (3.1). Fix $i \in \{1, 2, \ldots, p\}$ and $x \in X$ arbitrarily. Then, the dual problem of

$$\begin{array}{ll} \text{maximize} & g(x, \bar{t}^i) + \nabla_t g(x, \bar{t}^i)^\top (t - \bar{t}^i) - \frac{L}{2}\|t - \bar{t}^i\|^2 \\ \text{subject to} & t \in T = \{t \in \mathbb{R}^m \mid At \le b\} \end{array} \tag{5.2}$$

can be represented as

$$\begin{array}{ll} \underset{\eta\in\mathbb{R}^l}{\text{minimize}} & \frac{1}{2L}\|q_{\bar{t}^i}(x, \eta)\|^2 - r_{\bar{t}^i}(x, \eta) \\ \text{subject to} & \eta \ge 0, \end{array} \tag{5.3}$$

where,

$$\begin{aligned} q_{\bar{t}^i}(x, \eta) &:= -L\bar{t}^i - \nabla_t g(x, \bar{t}^i) + A^\top \eta, \\ r_{\bar{t}^i}(x, \eta) &:= \nabla_t g(x, \bar{t}^i)^\top \bar{t}^i + \frac{L}{2}\|\bar{t}^i\|^2 - g(x, \bar{t}^i) - b^\top \eta. \end{aligned}$$

[2, Section 5.2.4]. Since the strong duality holds between (5.2) and (5.3), NLP(E) can be rewritten equivalently as

$$\begin{aligned} &\underset{x}{\text{minimize}} && f(x) \\ &\text{subject to} && \min\left\{\frac{1}{2L}\|q_{\bar{t}^i}(x,\eta)\|^2 - r_{\bar{t}^i}(x,\eta) | \eta \geq 0\right\} \leq 0 \ (i = 1, 2, \ldots, p), \end{aligned}$$

which is also equivalent to the following optimization problem with a finite number of constraints:

$$\begin{aligned} &\underset{x,\eta^1,\ldots,\eta^p}{\text{minimize}} && f(x) \\ &\text{subject to} && \frac{1}{2L}\|q_{\bar{t}^i}(x,\eta^i)\|^2 - r_{\bar{t}^i}(x,\eta^i) \leq 0, \eta^i \geq 0 \ (i = 1, 2, \ldots, p). \end{aligned}$$

This is a convex programming problem with convex quadratic constraints if $g(\cdot, t)$ is affine. Hence, it can be solved effectively by means of the interior point method.

5.2 How to determine the Lipschitz constant

Although Algorithm 3.1 requires the Lipschitz constant L satisfying (3.3), it is not easy to find it in general. Alternatively, we set the value of $L_{\bar{t}}$ for each $\bar{t} \in E^r$ respectively and increase the value of $L_{\bar{t}}$ in each inner iteration so that the following inequality holds for each $\bar{t} \in E^r$:

$$g(v^r, \bar{t}) \leq g\left(v^r, P_T\left(\bar{t} + \frac{1}{L_{\bar{t}}}\nabla_t g(v^r, \bar{t})\right)\right). \tag{5.4}$$

Notice that (5.4) automatically holds when $L_{\bar{t}}$ ($\bar{t} \in E^r$) is the Lipschitz constant since we have

$$g(v^r, \bar{t}) \leq \max_{t \in T}\left\{g(v^r, \bar{t}) + \nabla_t g(v^r, \bar{t})^\top (t - \bar{t}) - \frac{L_{\bar{t}}}{2}\|t - \bar{t}\|^2\right\}$$

and

$$\max_{t \in T}\left\{g(v^r, \bar{t}) + \nabla_t g(v^r, \bar{t})^\top (t - \bar{t}) - \frac{L_{\bar{t}}}{2}\|t - \bar{t}\|^2\right\} \leq g\left(v^r, P_T\left(\bar{t} + \frac{1}{L_{\bar{t}}}\nabla_t g(v^r, \bar{t})\right)\right),$$

where the second inequality follows from (3.7) with $E := E^r, x := v^r$ and $t := P_T(\bar{t} + \frac{1}{L_{\bar{t}}}\nabla_t g(v^r, \bar{t}))$. Note that, even if (5.4) is satisfied in each iteration, the output of the algorithm may not be the optimum of the original SIP, since the property (3.3) may not hold. In such a case, we may restart the algorithm with letting $L_{\bar{t}}$ be a larger value, and the initial index set T^0 be equal to E^r obtained in the final iteration.

6 Numerical experiments

In this section, we implement Algorithm 3.1 and report some numerical results. The program is coded in Matlab 7.4.0(R2007a) and run on a machine with an Inter(R) Core(TM)2 Duo E6850 3.00GHz CPU and 3GB RAM. For the sake of comparison, we also implement another exchange-type method named Exchange 2, in which we update the finite index set as $\overline{E}^{r+1} = \overline{E}^{r} \cup \{t^r_{\mathrm{new}}\}$ and solve a sequence of finitely relaxed problems (3.2) instead of NLP(E) in Step 1.

Test problem

As a test problem, we consider a semi-infinite program derived from the Chebyshev approximation problem. Given a function $h : \mathbb{R} \to \mathbb{R}$, one of the typical Chebyshev approximation problem is to determine the coefficients $(x_1, x_2, \ldots, x_{n-1})^\top \in \mathbb{R}^{n-1}$ such that $\sum_{i=1}^{n-1} x_i t^i \approx h(t)$ over a compact set $T(\subseteq \mathbb{R})$, where t^i denotes the i-th power of $t \in \mathbb{R}$. This can be naturally reformulated as

$$\min_{x\in\mathbb{R}^n} \max_{t\in T} \left| h(t) - \sum_{i=1}^{n-1} x_i t^i \right| .$$

By using an auxiliary variable $x_n \in \mathbb{R}$, the above problem can be transformed into the following semi-infinite program with two linear semi-infinite constrains:

$$\begin{aligned} \underset{(x_1,x_2,\ldots,x_n)^\top\in\mathbb{R}^n}{\text{minimize}} \quad & x_n \\ \text{subject to} \quad & \sum_{i=1}^{n-1} x_i t^{i-1} - h(t) \le x_n \ (t \in T), \\ & -\sum_{i=1}^{n-1} x_i t^{i-1} + h(t) \le x_n \ (t \in T). \end{aligned} \tag{6.1}$$

In the experiment, we actually solve the above SIP with the following specific data:

$$n = 9,\ T = [-5, 5],$$

and

$$h(t) = \begin{cases} t + \frac{5}{6}\pi & (t \le \frac{5}{6}\pi), \\ \sin(t + \frac{5}{6}\pi) & (-\frac{5}{6}\pi < t \le 0), \\ \frac{1}{2}(1 + \sqrt{3} - \sqrt{3}\exp(t)) & (0 < t \le 2), \\ 5t^2 - \frac{1}{2}(40 + \sqrt{3}\exp(2))t + \frac{1}{2}(1 + \sqrt{3} + \sqrt{3}\exp(2)) & (t > 2). \end{cases}$$

Details of implementation

The actual implementation of Algorithm 3.1 and Exchange 2 are carried out as follows. In Step 0, we set the initial index set T^0 as $T^0 = \{-5 + \frac{5}{4}q\}_{q=0,1,\dots,8}$. In Steps 1-0 and 1-2 of Algorithm 3.1, we solve NLP(E) of the form (5.1) with $P_T(s) := \mathrm{med}(-5, s, 5)$. For solving NLP$(E)$ and the finite relaxed problem (3.2), we make use of *fmincon* solver in Matlab Optimization Toolbox. In Step 1-1, if we cannot find t^r_{new} satisfying $g(v^r, t^r_{\mathrm{new}}) > \gamma_k$, then we have to check the nonnegativity of $\max_{t\in T}(g(v^r, t) - \gamma_k)$. For solving $\max_{t\in T} g(v^r, t)$, we first choose grid points $\bar{t}_i := -5 + (i-1)/1000$ $(i = 1, \dots, 10001)$ from the index set T and let $t_{\max} \in \mathrm{argmax}_{1\le i\le 10001}\, g(v^r, \bar{t}_i)$. Then we further set 10000 grid points in $[t_{\max} - 1/10000, t_{\max} + 1/10000]$, and find the maximum of $g(v^r, t)$ for those 10000 points.

We next explain how to determine the value of L in NLP(E_r). Let $E_r \subseteq T$ be an arbitrary finite set given as in (3.1). Fix $i \in \{1, 2, \dots, p\}$ and $x \in X$ arbitrarily. Let $\{L^r_i\}_{r\ge -1}$ $(i = 1, 2, \dots, p)$ be a sequence of positive numbers satisfying $L^{r+1}_i = 2^{p^r_i} L^r_i$ where $p^r_i \ge 0$ is the smallest integer such that

$$g(v^r, \bar{t}^i) \le g\left(v^r, P_T\left(\bar{t}^i + \frac{1}{2^{p^r_i} L^r_i}\nabla_t g(v^r, \bar{t}^i)\right)\right) \tag{6.2}$$

for each $\bar{t}^i \in E_r$. We then set $L = L^r_i$ $(i = 1, 2, \dots, p)$ in the r-th iteration of Step 1.

Experiment 1

In the first experiment, we set $\gamma_0 = 10^{-5}, t^r_{\mathrm{new}} \in \mathrm{argmax}_{t\in T}\, g(v^r, t)$ and various value of L_i^{-1} $(i = 1, 2, \dots, 9)$ defined in (6.2). Then we run Exchange 2 and Algorithm 3.1 for solving SIP(6.1). The obtained results are shown in Table 1 and Table 2 where

optval:	the objective functional value of SIP(6.1) in the final iteration;
$\max g$:	the value of $\max_{t\in T} g(v^r, t)$ in the final iteration;
iter:	the number of inner iterations in Step 1;
time(sec):	computational time in seconds;
Algorithm 3.1(M):	Algorithm 3.1 with $L_i^{-1} = M$ for all i.
T_{fin}:	the r-th index set E_r in the final iteration.

We also give Figures 1 and 2 showing how the objective functional value for SIP(6.1) and $\max_{t\in T} g(v^r, t)$ vary as the inner iteration proceeds in the exchange method and Algorithm 3.1 with $L_i^{-1} = 20$ for all i. From the tables, we can observe that Exchange 2 and Algorithm 3.1 with $L_i^{-1} = 20$ and 100 find an optimum of SIP (6.1) successfully[2]. However, Algorithm 3.1 with $L_i^{-1} = 10$ fails to attain the optimum although it obtains a feasible point such that $\max_{t\in T} g(v^r, t) = 8.01 \times 10^{-7}$. This fact means that the value of L is smaller than the value of Lipschitz constant. From Figs. 1 and 2, we can also observe that Algorithm 3.1 finds the optimal solution in a lower number of iterations than Exchange 2. This may represent that NLP(E_r) approximates SIP(6.1) more precisely than the finite relaxed problem (3.2).

Algorithm	*optval*	$\max g$	time(sec)	iter
Exchange 2	0.465	1.58×10^{-6}	17.5	20
Algorithm 3.1(20)	0.465	2.26×10^{-6}	14.6	16
Algorithm 3.1(10)	0.504	8.01×10^{-7}	9.33	10
Algorithm 3.1(100)	0.465	7.07×10^{-6}	16.6	18

Table 1: comparison of the exchange method and Algorithm 3.1 (Experiment 1)

Algorithm	T_{fin}
Exchange 2	$\{-4.56, -3.29, -1.57, 0.150, 0.153, 1.59, 2.41, 3.59, 4.61, 5\}$
Algorithm 3.1(20)	$\{-4.56, -3.29, -1.57, 0.154, 1.59, 2.41, 3.60, 4.61, 5\}$
Algorithm 3.1(10)	$\{-4.85, -3.53, -1.70, 0.093, 1.58, 2.44, 3.69, 5\}$
Algorithm 3.1(100)	$\{-4.56, -3.30, -1.57, 0.150, 1.59, 2.41, 3.59, 4.61, 5\}$

Table 2: the index set obtained for the two methods (Experiment 1)

[2] We can observe that the regular exchange method can find the optimal value since the output solution obtained by the regular exchange method satisfies the feasibility of SIP fully. Then, we can observe that Algorithm 3.1(20,100) also finds the optimal solution since the output solution obtained by Algorithm 3.1(20,100) is feasible and the optimal value obtained by Algorithm 3.1(20,100) coincides the optimal value obtained by the exchange method.

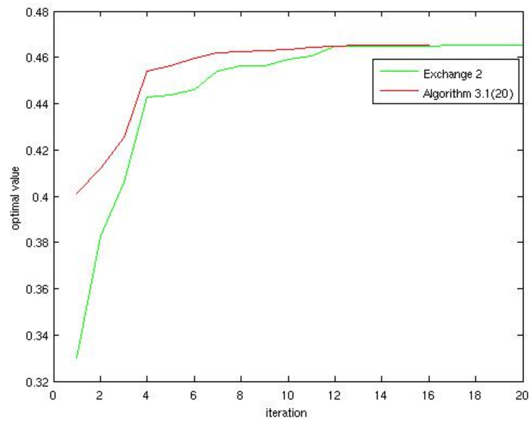

Fig. 1: the optimal value for two methods

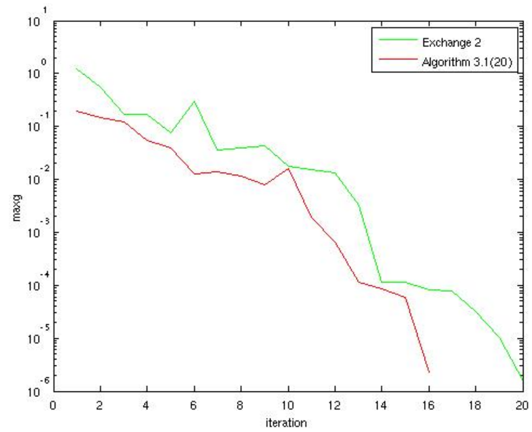

Fig. 2: $\max g$ for two methods

Experiment 2

In Experiment 1, we choose t^r_{new} such that $t^r_{\text{new}} \in \text{argmax}_{t\in T}\, g(v^r, t)$. However, to solve $\max_{t\in T} g(v^r, t)$, it often requires a high computational cost especially when T has a high dimension. In this experiment, we consider other choices of t^r_{new}. Specifically, we select t^r_{new} as follows: we first choose N grid points $\bar{t}_i = -5 + 10i/N$ $(i = 0, 1, \ldots, N)$ from the index set T and compute $g(v^r, t)$ for $t = \bar{t}_1, \bar{t}_2, \ldots, \bar{t}_N \in T$. If we find a $\bar{t} \in \{\bar{t}_1, \bar{t}_2, \ldots, \bar{t}_N\}$ such that $g(v^r, \bar{t}) > \gamma_k$, then we set $t^r_{\text{new}} := \bar{t}$. We run Algorithm 3.1 and Exchange 2 with $L_i^{-1} = 20$ for all i and various choices of N. The results are shown in Tables 3 and 4 where

Algorithm 3.1($N = M$): Algorithm 3.1 with $N = M$;
Exchange 2($N = M$): Exchange 2 with $N = M$.

We also give Figures 3 and 4 showing how the objective function value for SIP (6.1) and $\max_{t \in T} g(v^r, t)$ vary as the inner iteration proceeds in Exchange 2 and Algorithm 3.1 with $L_i^{-1} = 20$. From the tables, we can observe that Algorithm 3.1 with $N = 1000, 10$ and 100 succeed in obtaining an optimum of SIP (6.1). On the other hand, Exchange 2 cannot find an optimum within the time limit. Also, from the figures, we can observe that Algorithm 3.1 converges rapidly at first, but it converges slowly from the middle.

Algorithm	*optval*	$\max g$	time(sec)	iter
Exchange 2($N = 100$)	0.465	4.97×10^{-3}	fail	fail
Algorithm 3.1($N = 1000$)	0.465	7.97×10^{-6}	16.0	18
Algorithm 3.1($N = 100$)	0.465	9.05×10^{-6}	56.2	67
Algorithm 3.1($N = 10$)	0.465	9.69×10^{-6}	89.0	106

Table 3: comparison of the exchange method and Algorithm 3.1 (Experiment 2)

Algorithm	T_{fin}
Exchange 2($N = 100$)	$\{-4.6, -3.3, -1.6, 0.2, 1.6, 2.4, 3.6, 4.6, 5\}$
Algorithm 3.1($N = 1000$)	$\{-4.56, -3.29, -1.57, 0.151, 1.59, 2.41, 3.60, 4.61, 5\}$
Algorithm 3.1($N = 100$)	$\{-4.56, -3.29, -1.57, 0.157, 1.59, 2.41, 3.59, 4.61, 5\}$
Algorithm 3.1($N = 10$)	$\{-4.56, -3.29, -1.57, 0.150, 1.59, 2.41, 3.59, 4.61, 5\}$

Table 4: the index set obtained for the two methods (Experiment 2)

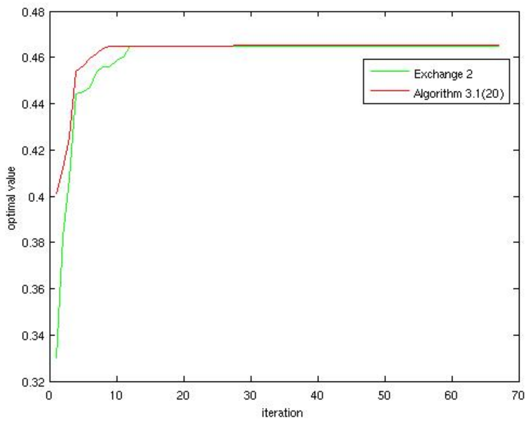

Fig. 3: the optimal value for two methods

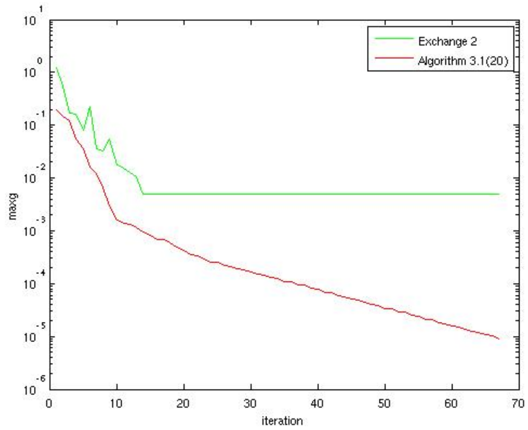

Fig. 4: $\max g$ for two methods

7 Conclusion

In this paper, we proposed the new algorithm for solving semi-infinite programming problems, and showed its convergence property under some assumptions. We also applied the algorithm to certain Chebyshev approximation problems and observed that the algorithm finds the SIP optimum efficiently. However, there still remain some future works. First, it is desired to relax the assumption that were used for the convergence analysis. Also, it is important to consider better techniques of how to choose the constant

L when the Lipschitz constant is unknown.

Acknowledgments

First of all, I would like to express my sincere appreciation to Associate Professor Nobuo Yamashita for his useful advice. Although I sometimes troubled to him by my faults, he always helped me kindly. I would like to tender my acknowledgment to Shunsuke Hayashi and Takayuki Okuno. They corrected my thesis and gave me helpful comments. I am also grateful to Assistant Professor Ellen Hidemi Fukuda for her numerous comments. Finally, I would like to thank all members of Yamashita Laboratory, my friends and my family for their tremendous support.

References

[1] B. Betro, An accelerated central cutting plane algorithm for linear semi-infinite programming, Mathematical Programming, 101 (2004), pp. 479–495.

[2] S. Boyd and L. Vandenberghe, Convex Optimization, Cambridge University Press (2004).

[3] F. Facchinel and J. S. Pang, Finite-Dimensional Variational Inequalities and Complementarity Problems, Vol. 2, Springer, (2003).

[4] S. Hayashi and S. Y. Wu, An explicit exchange algorithm for linear semi-infinite programming problems with second-order cone constraints, SIAM Journal on Optimization, 20 (2009), pp. 1527–1546.

[5] K. O. Kortanek and H. No, A central cutting plane algorithm for convex semi-infinite programming problems, SIAM Journal on Optimization, 3 (1993), pp. 901–918.

[6] T. Okuno, S. Hayashi and M. Fukushima, A regularized explicit exchange method for semi-infinite programs with an infinite number of conic constraints, SIAM Journal on Optimization, 22 (2012), pp. 1009–1028.

[7] E. Polak, Optimization: Algorithms and Consistent Approximations, Springer, (1997).

[8] R.T. Rockafellar, Convex Analysis, Princeton University Press (1970).

[9] A. Shapiro, Semi-infinite programming, duality, discretization and optimality conditions, Optimization 58 (2009), pp. 133–161.

[10] G. Still, Discretization in semi-infinite programming: the rate of convergence, Mathematical Programming 91 (2001), pp. 53–69.

[11] K. L. Teo and X. Q. Yang and L. S. Jennings, Computational discretization algorithms for functional inequality constrained optimization, Annals of Operations Research 28 (2000), pp. 215–234.

Master's Thesis

An exchange method with refined subproblems for convex semi-infinite programming problems

Guidance

Associate Professor Nobuo YAMASHITA

Kensuke GOMOTO

Department of Applied Mathematics and Physics

Graduate School of Informatics

Kyoto University

February 2014

An exchange method with refined subproblems for convex semi-infinite programming problems	Kensuke GOMOTO	February 2014

An exchange method with refined subproblems for convex semi-infinite programming problems

Kensuke GOMOTO

Abstract

The semi-infinite programming problem (SIP) is an optimization problem with an infinite number of constraints in a finite dimensional space. The SIP has been studied extensively so far, since a lot of practical problems in various fields such as physics, economics, and engineering can be formulated as the SIPs. The exchange method is one of the most useful algorithms for solving the SIP, and it has been developed by many researchers. In this paper, we focus on the convex SIPs and propose a new exchange method for solving them. While the traditional exchange method solves a sequence of the relaxed problems with finitely many constraints that are selected from the original constraints, our method solves a sequence of semi-infinite programs relaxing the original SIP. These relaxed problems can be solved efficiently by transforming them into certain optimization problems with finitely many constraints. Moreover, under some mild assumptions, they approximate the original SIP more precisely than the finite relaxed problems in the traditional exchange method. We also establish global convergence of the proposed method under strict convexity assumption on the objective function, and examine its efficiency through some numerical experiences.